AD-A120 343 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/4
CARGO LOADING--A PROPOSED APPROACH FOR MAXIMIZING SPACE UTILIZA--ETC(U)
SEP 82 G GERSTEL
UNCLASSIFIED DTNSRDC-82/099 NL

1 OF 2
AD A
120343

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC-82/099	2. GOVT ACCESSION NO. AD-A120343	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CARGO LOADING--A PROPOSED APPROACH FOR MAXIMIZING SPACE UTILIZATION OF CONTAINERS LOADED WITH PALLETIZED LOADS		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gerald Gerstel		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (See reverse side)
11. CONTROLLING OFFICE NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084 (Code 187)		12. REPORT DATE September 1982 13. NUMBER OF PAGES 169
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) H. Lieberman Code SUP033		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Containerization
Cargo Containers
Palletized Loads
Packing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An in-depth analysis of space utilization in cargo containers to be loaded with palletized loads and shipped by barge indicates that space utilization can be significantly improved. This report describes a loading procedure which uses analytic techniques to precisely define the amount and arrangement of loads at the three different loading levels involved. Development of a computer program to implement the results of the study is recommended.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Block 10)

Program Element 62760N
Task Area TF 60 531 005
Work Unit 1800-007

TABLE OF CONTENTS

	Page
LIST OF FIGURES	v
LIST OF TABLES	vi
ABSTRACT	1
ADMINISTRATIVE INFORMATION	1
INTRODUCTION	3
ALTERNATIVE APPROACHES TO AUTOMATED LOADING	5
FIXED PATTERN APPROACH	5
ANALYTIC FORMULATIONS	7
PROPOSED APPROACH	11
OVERVIEW	11
GUIDELINES FOR SOLUTION STRATEGY	12
INPUT REQUIREMENTS AND PRELIMINARY LOAD DATA ANALYSIS	25
PALLET LOADING	28
Pattern Formation	28
Stacking	47
Completion	57
CONTAINER LOADING	59
BARGE LOADING	61
ORGANIZATION AND FUNCTIONAL DESCRIPTION OF PROPOSED PROGRAM LOADER	63
SAMPLE PROBLEM	65
STATEMENT OF PROBLEM	65
SOLUTION	67
Assumptions and Restrictions	67
Preliminary Load Data Analysis	68
Pallet Loading	70
Group 1 Processing	71
Group 2 Processing	73
Group 3 Processing	78
Group 4 Processing	81
Group 5 Processing	84
Group 6 Processing	85
Group 7 Processing	86
Group 8 Processing	87
Remainder Group Processing	88

Page

Stacking and Completion 89

<u>Container #1</u> 90

<u>Final Stack #1</u> 94

<u>Final Stack #2</u> 95

<u>Container #2</u> 96

<u>Initial Stack #1</u> 97

<u>Initial Stack #2</u> 99

Container Loading 102

DISCUSSION OF RESULTS 102

RECOMMENDATIONS AND CONCLUSIONS 105

ACKNOWLEDGMENTS 109

REFERENCES 111

APPENDIX A - INDEX CHART FOR PALLET PATTERN DETERMINATION 113

APPENDIX B - PALLET PATTERNS ON 40" X 48" PALLETS 117

APPENDIX C - FLOWCHARTS OF PROPOSED PROGRAM LOADER 121

LIST OF FIGURES

	Page
1 - Difficulty of Mixing Box Sizes Using Fixed Patterns	6
2 - Pictorial Overview of Proposed Approach	13
3 - Example of Cluster Types	30
4 - Type 1 Configuration	36
5 - Type 2 Configuration	38
6 - Type 3 Configuration	40
7 - Type 4 Configuration	43
8 - Pattern Completion by Nesting of Type 3 and Type 4 Configurations	45
9 - Alternative Pallet Loads Spanning Identical Stack Height	50
10 - Improvement of Pallet Load Stability	60
11 - Example of Pattern Completion by Synthesizing Group Height	107
12 - Generalization of Row, Column Construction Techniques	107
13 - Main Program LOADER, Flowchart	121
14 - Subroutine INPUT, Flowchart	122
15 - Subroutine ANALYZ, Flowchart	123
16 - Subroutine PATRNS, Flowchart	124
17 - Subroutine STRING, Flowchart	130
18 - Subroutine TYPE1, Flowchart	133
19 - Subroutine TYPE2, Flowchart	134
20 - Subroutine TYPE3, Flowchart	135
21 - Subroutine TYPE4, Flowchart	138

	Page
22 - Subroutine STACKS, Flowchart	140
23 - Subroutine STATE, Flowchart	145
24 - Subroutine PILES, Flowchart	146
25 - Subroutine PALLETS, Flowchart	148
26 - Subroutine STAKER, Flowchart	150
27 - Subroutine SUPPLY, Flowchart	155
28 - Subroutine UNLOAD, Flowchart	156
29 - Subroutine FILL, Flowchart	157
30 - Subroutine TRADE, Flowchart	158
31 - Subroutine CNTNER, Flowchart	159
32 - Subroutine BARGE, Flowchart	160

LIST OF TABLES

1 - System State Definitions	52
2 - State Transition Matrix	52
3 - Sample Problem Load Population Data	69

ABSTRACT

An in-depth analysis of space utilization in cargo containers to be loaded with palletized loads and shipped by barge indicates that space utilization can be significantly improved. This report describes a loading procedure which uses analytic techniques to precisely define the amount and arrangement of loads at the three different loading levels involved. Development of a computer program to implement the results of the study is recommended.

ADMINISTRATIVE INFORMATION

This research was performed for the Logistics Division, Code 187, of the Computation, Mathematics and Logistics Department, Code 18. The author was on the staff of the Computer Aided Design and Manufacturing Division, Code 185. The work was sponsored by the Naval Supply Systems Command, Code 043 as a subtask of Work Request WR-09012, Program Element 62760N, Task Area TF 60 531 005, and Work Unit 1800-008.

INTRODUCTION

One of the foremost problems in the field of containerization is the need for improving the use of cubic capacity of cargo containers. Both Tabak[1*] and NAVSEASYSCOM[2] mention the significance of the cube limitation problem and the inefficient manner in which storage space is used.

The importance of the problem and the economic savings to be realized if a successful solution could be obtained led to a study of space utilization in cargo containers to be loaded with palletized loads. Loading takes place at the three different levels, and each level affects the overall efficiency of space utilization. The three levels are:

- loading of boxes onto pallets
- loading of pallets into containers
- loading of containers onto a barge.

The results of the study, reported here, include the development of an automated loading procedure that precisely defines the amount and arrangement of loads at each level of loading.

---

*A complete listing of references is given on page 111.

ALTERNATIVE APPROACHES TO AUTOMATED LOADING

FIXED PATTERN APPROACH

MIL-STD-147B[3] describes and illustrates the practices and procedures for palletizing unit loads. Appendixes C and D of the standard, included in this report as Appendixes A and B, provide an important step in the direction of automated loading. The first appendix is an index chart for pallet pattern determination. The chart covers box sizes ranging in length from 6 to 52 inches in increments of half an inch, and in width from 6 to 43 inches in increments of half an inch. A pattern number read from the chart for a box of specified length and width corresponds to a pattern in the second appendix. Each of the 124 pallet patterns provides for efficient use of at least 80 percent of the pallet surface. However, there are numerous box sizes for which no pattern is provided (for example, 15"W X 18"L), and generally a pattern can be relied upon only if all boxes are the same size. For example, uniform loads of boxes 17-1/2"W X 34"L or of boxes 21-1/2"W X 28"L can be loaded on a 40" X 48" pallet using Pattern 1 as shown in Figures 1A and 1B. However, if the sizes are mixed as shown in Figure 1C, the pattern is not suitable, since the maximum allowable load width is 52 inches. If the boxes are arranged as shown in Figure 1D, the pattern is acceptable.

The idea of using previously developed, efficient, fixed patterns for selected box sizes is simple and logical, but has several limitations:

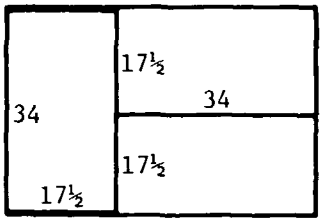


Figure 1a - Acceptable loading of 17½" x 34" boxes using pattern 1

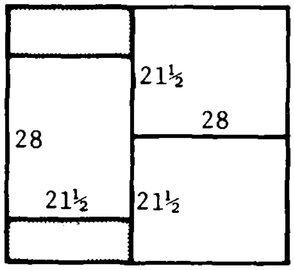


Figure 1b - Acceptable loading of 21½" x 28" boxes using pattern 1

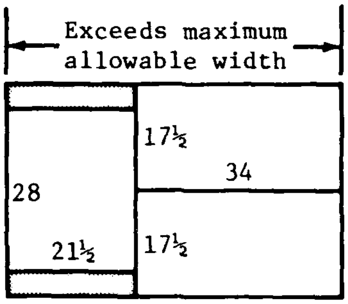


Figure 1c - Unacceptable loading of 2 sizes of boxes using pattern 1

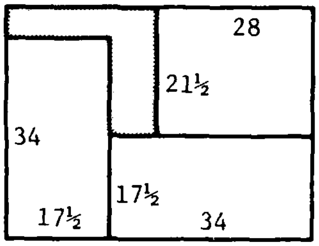


Figure 1d - Acceptable loading of 2 sizes of boxes using pattern 1

Figure 1 - Difficulty of Mixing Box Sizes Using Fixed Patterns

- If the number of permissible box sizes is large, the number of pre-developed fixed patterns becomes large; if mixing of sizes is permitted, determining the large number of mixed patterns becomes infeasible.

- There is no correspondence between the actual frequency of use of certain sizes and the frequency with which the sizes occur in the fixed patterns.

- There is no mechanism for improving space utilization efficiency; if a fixed pattern is selected with a reasonable amount of waste, a more efficient pattern might be easily found but is never sought.

- If the allowable load widths or lengths are changed, or if the maximum permissible dimensions are variable, then considerable rework may be needed to expand or alter the fixed patterns. The same difficulties would apply if the permissible box sizes were altered.

ANALYTIC FORMULATIONS

A more general approach than using fixed patterns would be to allow for linear combinations of box sizes and more general types of permissible pattern variations. Gilmore and Gomory,[4] Herz[5] and Christofides and Whitlock[6] reported work of this nature but in a different context.

Gilmore and Gomory[4] consider multi-stage cutting stock problems of two and more dimensions. In the two-dimensional cutting stock problem a supply of stock rectangles of width W and length L is used to fill a demand for N_i rectangles of width w_i and length l_i, $i = 1,2,...m$. The

problem is to cut the stock rectangles into smaller demand rectangles using as few stock rectangles as possible. In the cargo loading problem, the fixed size pallet corresponds to the stock rectangle of the cutting stock problem and the width and length of the box correspond to the demand rectangle dimensions. The number of boxes of each size corresponds to the N_i rectangles and the number of different size boxes corresponds to the m rectangles. In both problems, the objective is to minimize waste. In the three-dimensional problem, a rectangular parallelepiped is used instead of a rectangle, and its height becomes the third dimension.

Gilmore and Gomory[4] discuss a linear programming formulation and consider the computational difficulties which arise from the immense number of columns that can occur in the matrix. For the one-dimensional cutting stock problem, the computational difficulty can be reduced by transforming the problem and solving it as a knapsack problem. Unfortunately, for two dimensions, the analagous transformation from the linear programming problem into a generalized knapsack problem results in a problem for which no practical means of solution is known. The difficulty increases if the three-dimensional problem is considered.

Herz[5] considers the two-dimensional cutting stock problem. The Gilmore-Gomory solution technique involves an iterative algorithm, but the Herz algorithm is recursive and its author claims that, when implemented, it results in higher computational speeds in the solution of the cutting-stock problem. Herz gives some computational results for a program which was written in PL/1 and executed on an IBM computer.

Christofides and Whitlock present a tree-search algorithm in which the maximum number of each type of piece produced is constrained. A dynamic programming solution technique is used. The results of a program developed for the CDC-7600 computer indicate that the algorithm is effective in solving cutting-stock problems of moderate size. When ten pieces are to be cut from the stock rectangles, the solution time is approximately 15 seconds. When the number of pieces is increased to twenty, the solution time is increased to 1 minute. For a larger number of pieces, the number of nodes in the tree and computational times become so large as to be impractical.

PROPOSED APPROACH

OVERVIEW

The approach proposed here uses analytic procedures to generalize the fixed pattern approach and to take into account the specific loading constraints associated with the cargo loading problem. This approach necessitates the development of a computer program which permits the three-dimensional cargo loading problem to be attacked from a more realistic point of view. The key concepts involve:

- Iteration to improve space utilization to within acceptable tolerances; implicit exclusion of unwarranted iterations.
- Generation of load clusters.
- Generation of configuration types dependent on priorities and on frequency of box size occurrence within the load population.
- Pattern completion by integer linear combination and nesting of clusters.
- Generation of load stacks.
- Stack filling for crude but rapid utilization of space.
- Load trading for refinement of space utilization.
- Automatic insertion of pallets into load stacks for adjustment of pallet load heights.

- Automatic determination of arrangement of pallets in containers and containers in barges.

- Automatic monitoring of load weights.

- Automatic placement of loads for improved load stability.

- Ability to allow for variability in available box sizes, pallet sizes, containers sizes, barge types, and wasted space required for load accessibility.

- Ability to trade off space utilization efficiency versus computer costs.

- Provision for complete computer bookkeeping and output reporting, including space utilization efficiency, at all loading levels.

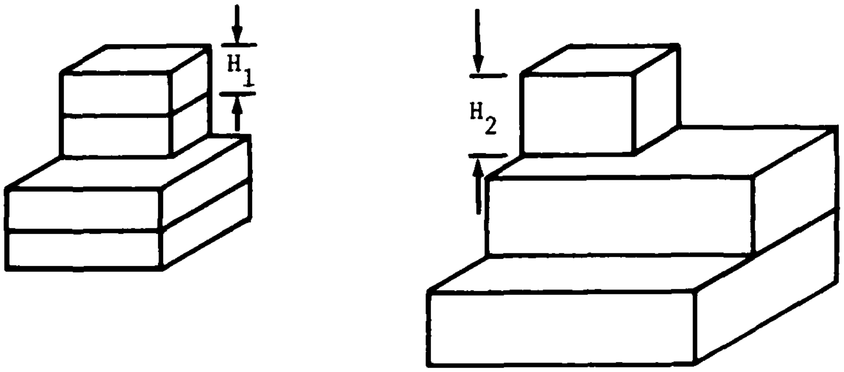
Although these concepts have been listed separately, many of them are interdependent and receive simultaneous consideration in the development of the problem solution. A pictorial overview of the proposed approach appears in Figure 2.

GUIDELINES FOR SOLUTION STRATEGY

A number of basic criteria and mathematical operations must be stated or developed in order to understand the solution strategy. These fundamental ideas provide the basis for the decision rules that deal with important questions frequently raised in the loading process, such questions as:

Figure 2 - Pictorial Overview of Proposed Approach

1. Group boxes on basis of box heights



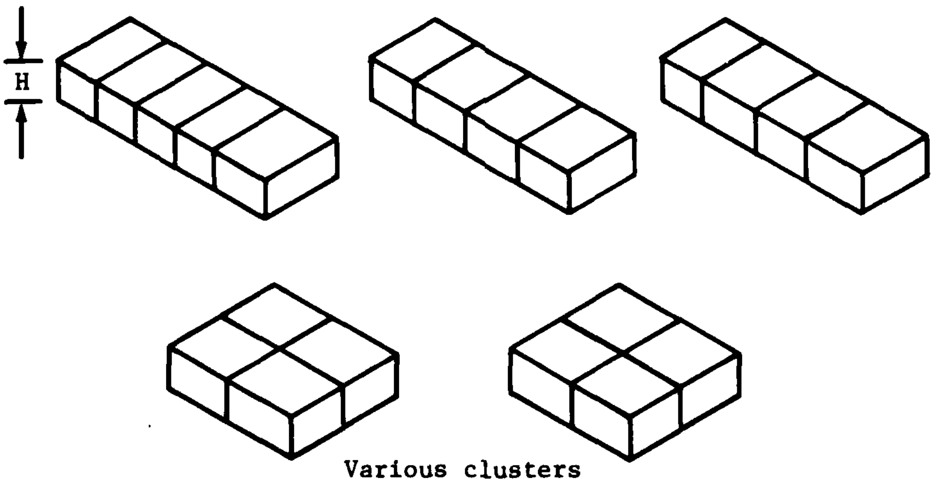
Group 1

All boxes of
height H in range
8" < H ≤ 10"

Group 2

All boxes of
height H in range
10" < H ≤ 12"

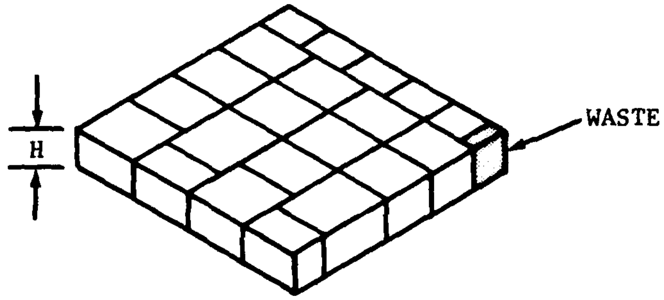
2. Generate box clusters within each group



Various clusters

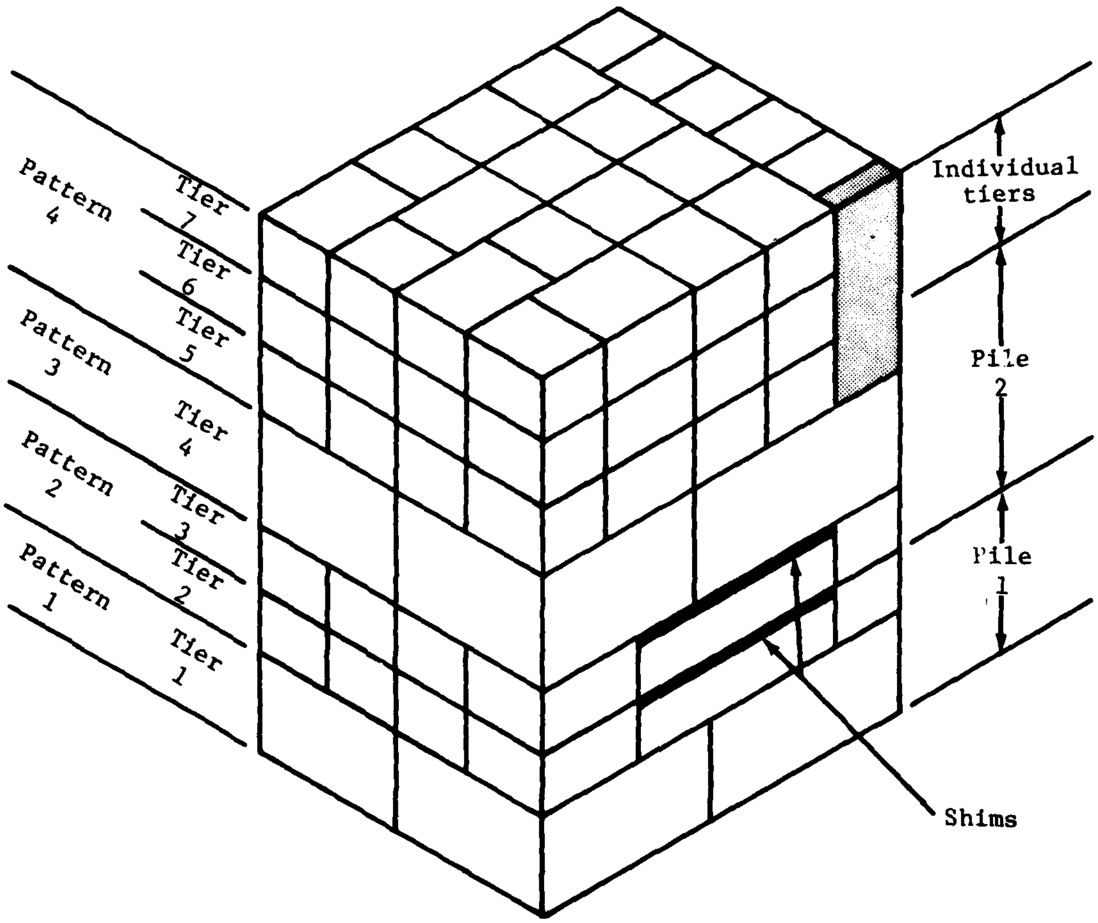
Figure 2 (Continued)

3. Form patterns by combining clusters



Pattern

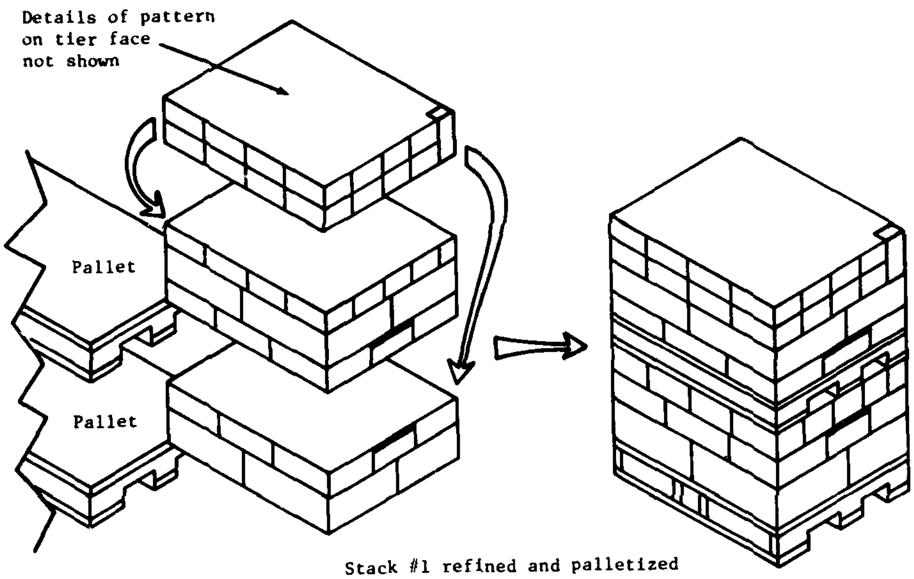
4. Assign patterns to tiers; and insert shims to level tiers; generate load stacks by stacking tiers into preliminary piles



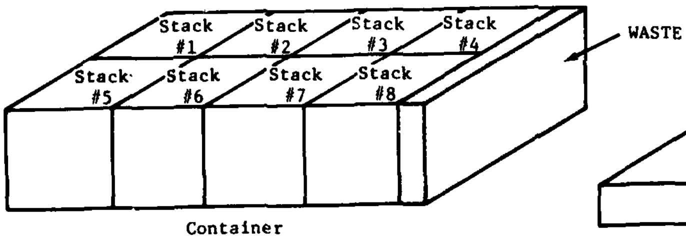
Stack #1

Figure 2 (Continued)

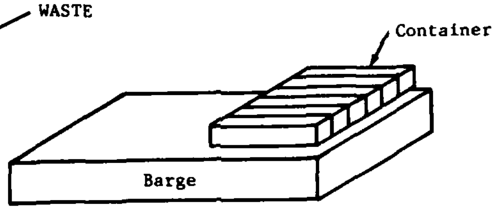
5. *Refine piles and insert pallets into stack*



6. Arrange stacks in container



7. Arrange containers on barge



- In what sequence should the boxes be selected from the load population?

- How should loading patterns be formed and what measure should be used to compare their effectiveness in space utilization?

- How can a current loading pattern be developed that gives consideration to enhancement of the load pattern opportunities for the remaining load population?

- If wasted space exists, what types of modifications should be allowed in altering the configuration and composition of a given subset of the load population to improve space utilization?

The first guideline to be introduced is associated with the utility or value of a loading pattern and the contributing worth of each of its components. Ideally, it would be desirable to have a precise quantitative measure of each of these factors, but unfortunately none is available. For a small load population, trial and error or possibly the use of dynamic programming might provide the best overall utilization of space and, a posteriori, assign the highest value of utility to this result.for a large population, this approach is not practical.

Intuitively, anyone concerned with a loading problem is inclined to place greater value on the satisfactory placement of a large object than a small one. This could be justified on the basis that a larger object, by definition, requires more space than a smaller one. It is important to maximize the opportunities for placement of a larger object; the number of opportunities is greater without the space limitations resulting from

prior placement of smaller objects. Alternatively, if a large object is placed first, the likelihood of finding an acceptable place for a follow-on object would be greater if the follow-on object were smaller. This intuitive idea is used for establishing the precedence guideline that, all other factors being equal, a larger box will always be selected and scheduled before a smaller one. The phrase, "all other factors being equal," is important. For example, if in certain instances it could be determined that selection of a smaller box in lieu of a larger one would ultimately result in less wasted space, then in this case, all other factors are not equal and the precedence relationship could be reversed. If, however, it could not be determined, a priori, which size selection would result in best overall s ace utilization, then the precedence guideline would be applicable and larger sizes would be assumed to have higher box size values.

Space utilization can be measured in terms of space utilization efficiency, which is defined as:

$$\text{Space Utilization Efficiency} = \frac{\text{Volume of Space Utilized}}{\text{Volume of Available Space}} \times 100\%$$

Although the objective is to maximize the space utilization efficiency at each phase of the loading process, sometimes it may be necessary to trade off space utilization efficiency against the value of the box sizes contained in a particular loading pattern. It may be advantageous to use for a large box a pattern having relatively high waste with the expectation of finding for a smaller size box a more efficient pattern

which will give greater space utilization efficiency for the complete load.

In order to trade off box size value versus space utilization efficiency, a tolerance must be specified to denote the amount of space acceptable as waste, provided that sufficient box size value is achieved.

A mathematical technique, known as the integer linear combination technique, has been developed which combines the box size value and space utilization concepts. This technique can be used to accelerate and make decisions concerning the load solution process. In applying the technique, it is assumed that the precedence guideline always applies to the first box size to be selected. After selection of the largest box, successive selections will still pick the same size box if it is still available. Therefore, it becomes advantageous to determine first the maximum number of the largest size box that will fit in the available space and at the same time produce an acceptable configuration with respect to minimum waste. The remaining load population is then checked to determine whether the quantity is available, and, if so, all boxes of that size are simultaneously selected. If either the configuration or size availability precludes the maximum quantity assignment, then the assignment is reduced to the maximum available quantity that produces an acceptable configuration.

The technique can be used to solve two problem versions which depend on whether two sizes of a box x_1 and x_2 are known or if only box size x_1 is known, and the other size x_2 is unknown. The integer combination

applies to a one-dimensional space, since the items to be combined will always have a common dimension, and their combination will always be sought in the remaining one-dimensional space.

Problem Version A

A one-dimensional space of magnitude X is to be filled within tolerance TOL by integer linear combination of specified size x_1 and unknown size x_2. The availability of x_1 and x_2 is given by constants n_{1AVAIL} and and n_{2AVAIL}. Find maximum n_1 that satisfies:

$$X - TOL \leq n_1x_1 + n_2x_2 \leq X$$

$$n_1 \leq n_{1AVAIL}$$

$$n_2 \leq n_{2AVAIL}$$

$$n_1 > 0$$

$$n_2 \geq 0$$

$$n_1, n_2 \text{ integer}$$

Problem Version B

A one-dimensional space of magnitude X is to be filled within tolerance TOL by integer linear combination of specified sizes x_1 and x_2. The availability of x_1 and x_2 is given by constants n_{1AVAIL} and n_{2AVAIL}.

Find maximum n_1 that satisfies:

$$X - TOL \leq n_1 x_1 + n_2 x_2 \leq X$$

$$n_1 \leq n_{1AVAIL}$$

$$n_2 \leq n_{2AVAIL}$$

$$n_1 > 0$$

$$n_2 \geq 0$$

$$n_1, n_2 \text{ integer}$$

<u>Integer Linear Combination Technique</u>

Problem Version A is solved:

Step 1: Set $n_2 = 0$ and $n_1 = \min \left\{ \left[\frac{X}{x_1} \right], n_{1AVAIL} \right\}$

where notation [ ] denotes smallest integer $\leq$ number in brackets.

Step 2: Test if n_1 satisfies $X - TOL \leq n_1 x_1 \leq X$. If relation holds, then n_1, n_2 are optimum. Otherwise, go to Step 3.

Step 3: Let x_{2SMALL} denote smallest available size of x_2 .

If x_{2SMALL} is void, go to Step 8.

Otherwise, set $n_2 = n_2 + 1$ and go to Step 4.

Step 4: Test if n_1, n_2 satisfy

$$\frac{X - n_1 x_1}{n_2} < X_{2SMALL}$$

where x_{2SMALL} denotes smallest available size.

If this relation holds, go to Step 7.

Otherwise, go to Step 5.

Step 5: Solve for x_{2LOWER} and x_{2UPPER}

where

$$x_{2LOWER} = \frac{X - XTOL - n_1 x_1}{n_2}$$

and

$$x_{2UPPER} = \frac{X - n_1 x_1}{n_2}$$

Let $\{x_2\}$ denote set of available sizes which satisfy

$x_{2LOWER} \leq x_2 \leq x_{2UPPER}$.

If $\{x_2\}$ is void, go to Step 3.

Otherwise, go to Step 6.

Step 6: Test if $n_2 \leq n_{2AVAIL}$ where n_{2AVAIL} denotes cumulative quantity of all members of $\{x_2\}$.

If relation holds, n_1 and n_2 are optimum and largest sizes belonging to $\{x_2\}$ are to be used.

Otherwise, go to Step 3.

Step 7: Set $n_1 = n_1 - 1$

and

$n_2 = 1$.

If $n_1 = 0$, go to Step 8.

Otherwise, go to Step 4.

Step 8: STOP! Problem has no solution.

Problem Version B is solved:

Step 1: Set $n_2 = 0$, $n_1 = \min \left\{ \left[\frac{X}{x_1}\right], n_{1AVAIL} \right\}$

$$n_{2MAX} = \min \left\{ \left[\frac{X}{x_2}\right], n_{2AVAIL} \right\}$$

Step 2: Test if $X - TOL \leq n_1 x_1 \leq X$. If it is, then n_1 and n_2 are optimum. Otherwise, go to Step 3.

Step 3: Set $n_2 = n_2 + 1$.

Test if $n_2 > n_{2MAX}$. If it is, go to Step 5.

Otherwise, go to Step 4.

Step 4: Test if $X - n_1 x_1 - n_2 x_2 < TOL$.

If relation holds, then n_1 and n_2 are optimum.

Otherwise, go to Step 3.

Step 5: Set $n_1 = n_1 - 1$ and $n_2 = 0$.

If $n_1 = 0$, go to Step 6.

Otherwise, go to Step 3.

Step 6: Stop! Problem has no solution.

The solution procedure ensures that unnecessary iterations are implicitly excluded when size availability becomes limited. For both problem versions, the iteration proceeds by initially allocating the maximum quantity to n_1 and the minimum to n_2. The procedure guarantees that n_1 and n_2 are integers.

In the process of generating and evaluating various loading patterns, criteria are needed to limit the types of loading patterns permitted. With sufficient time and financial expenditure, complicated, unusual loading patterns might be found to reduce wasted space; these solutions are ignored on the basis of practical considerations. The loading patterns sought are to be built up in a logical sequence of steps which favor uniformity of box size and regularity of configuration. The use of one size box is subject to its availability in the population or population subset, and also to the degree of success to be realized in the economical use of space. When it becomes necessary to mix sizes, usually a mix of only two sizes is permitted to keep the solution procedure simple and practical.

Regularity of configuration refers to configurations with parallel rows of boxes, parallel columns of boxes, and clusters of boxes.

INPUT REQUIREMENTS AND PRELIMINARY LOAD DATA ANALYSIS

To start the analysis, the user will input data on container size, pallet size, permissible pallet overhang, maximum allowable pallet loading height, batch size, barge type, various tolerances for different kinds of acceptable waste space, and, if applicable, reduced stack height for load accessibility.

The container size may be standard or non-standard. The solution technique will, in general, require that the user select the container size if more than one container size is available. The program will also be useful as a tool for selecting the container size by analytical repetition of the solution technique for various container sizes. The most efficient container size or sizes for the particular load on hand will be selected.

The pallet may be the normal 40-inch by 48-inch size or any other size. The permissible pallet overhang could extend the load size to as much as 43 inches by 52 inches for the normal pallet size. The program will have a stored table of overhang data extracted from MIL-STD-147B.[3]

The maximum allowable pallet loading height (including the pallet height) may be optionally specified by the user. Default values of 43 inches for SEAVAN and 41 inches for MILVAN as specified by MIL-STD-147B will be used.

The batch size is the maximum number of cargo boxes to be analyzed and schedul^d for loading by the program. Before actual program development, it is difficult to estimate what range of batch sizes should be specified to obtain both economical program execution and favorable space utilization. Tests will be required after program development to determine optimum batch size for specific loading problems.

The barge type is specified to identify the quantity and arrangement of available space for cargo stowage. The program will contain stored data for the LASH and SEABARGE to characterize the available space. All available space will be subdivided or, where necessary, approximated in terms of rectangular parallelepipeds. For barges other than LASH or SEABARGE, the user will be responsible for describing the available space.

The initial data will be followed by input which describes in detail the characteristics of the load. It is assumed that the load will consist of boxes of random sizes. The diversity of available sizes is exhibited in the Federal Supply Catalog.[7] For each box, the following information is needed:

- identification
- dimensions (specification of box cube instead of edge dimensions is unacceptable)

- permissible orientation
- weight
- destination
- special value (optional)

The last characteristic will be used by the program to alter the sequence in which the load scheduling steps occur.

Preliminary load data analysis is performed to arrange the data in the most convenient form for carrying out the loading process. Rearrangement of the data can reduce the number of iterations required to reach a solution and can also reduce computer execution costs. Additional advantages will become evident as the loading procedure progresses.

Each batch of data is categorized by box heights. A list of box groups is generated so that all boxes in a group will have a common height, or heights which are judged to be close enough on the basis of an acceptable tolerance. For a group in which all boxes do not have the same height, it is assumed that shims will be used and the group height will be considered the largest box height in the group.

The list of box groups is rearranged so that the groups are ordered by decreasing group heights. When boxes in different groups are compared, the value of any member of one group is considered to be proportional to

the group height, that is, a group of large height has greater value than one of a smaller height. Since all members within each group have, by definition, a common (or common within tolerance) height, area becomes the measure of value of a box and a decreasing sequence of box cross-sectional areas provides the yardstick for determining the next available box of maximum value.

PALLET LOADING

After preliminary load data analysis, the next step in problem solution is to load pallets with minimum waste. Pallet loading progresses in three stages. The first stage, pattern formation, is fundamental and is concerned with automatically forming efficient patterns for loading tiers of boxes. The stacking stage consists of analytically determining the sequence in which tiers are to be stacked. The completion stage involves automatic insertion of pallets into the stacks, additional checks and adjustments resulting from weight and stability considerations, and generation of pallet loading reports.

Pattern Formation

The difficulties of working with fixed patterns have already been noted. The development which follows attempts to provide flexibility in pattern formation and to correlate the remaining box populations in a batch with the types of patterns to be used.

The basic building blocks in pattern formation are referred to as clusters. A cluster is a particular configuration of boxes considered beneficial for pattern building. Each cluster is developed from the box of highest value consistent with the available space constraints. This box is referred to as the box seed, or just the seed. The following definitions differentiate among the types of clusters to be used. For brevity, parentheses denote additional definitions. Examples of various cluster types are given in Figure 3.

A row (column) cluster, or just a row (column), has boxes arranged in the form of a row (column).

A complete row (column) is one which can be placed on a pallet, parallel to a side, so that any resultant wasted space is small, that is, within tolerance. The width (height) of a complete row (column) must be equal to or only slightly less than the pallet width (height).

An incomplete row (column) is one having width (height) not within tolerance requirements for completeness.

A partial row (column) is one which fills within tolerance an available row (column) space of smaller width (length) than the pallet. The width (height) of a partial row (column) is too small for the row (column) to be designated as complete.

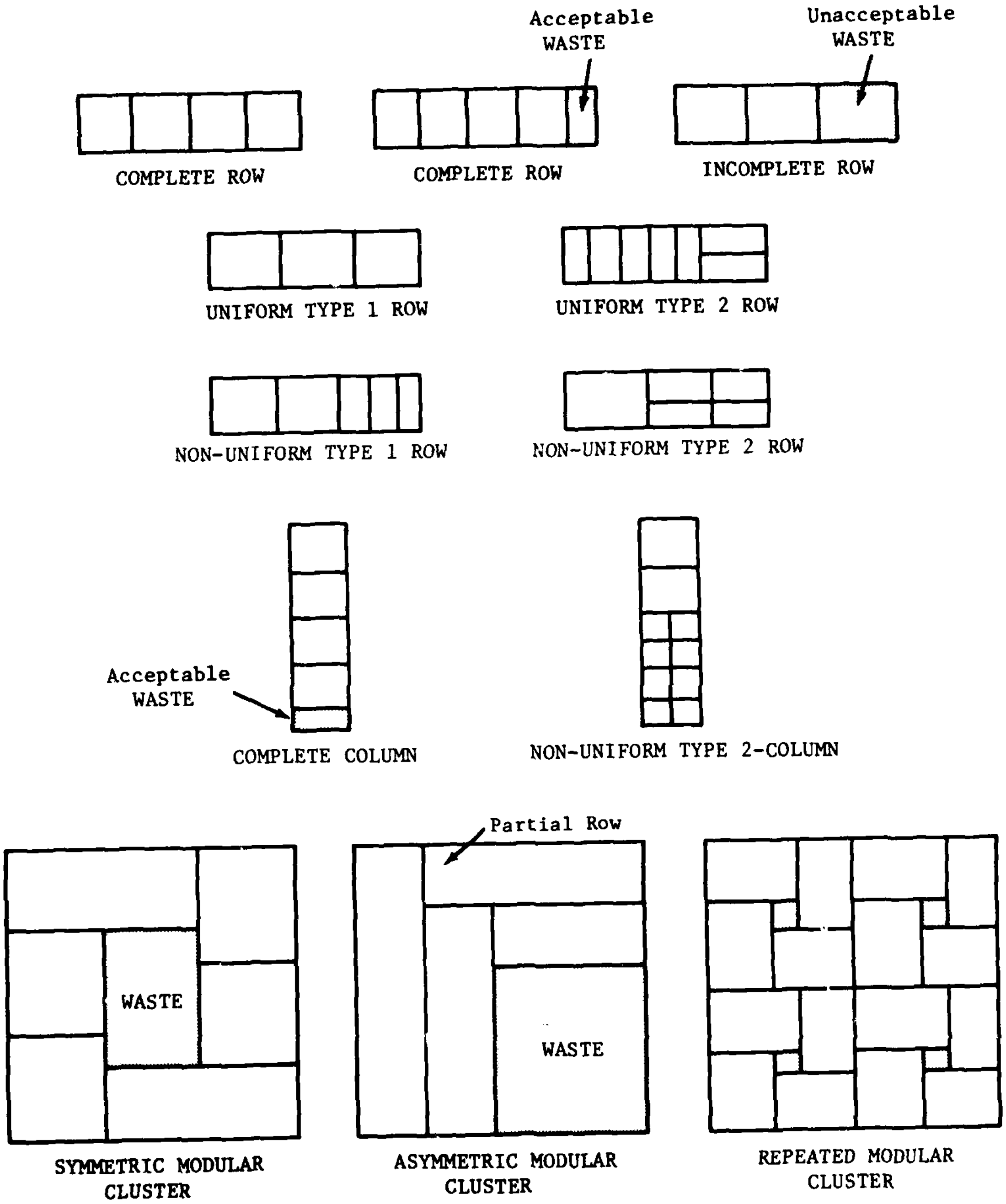


Figure 3 - Example of Cluster Types

A <u>uniform row</u> (<u>column</u>) is constructed from boxes of only one size.

A <u>non-uniform row</u> (<u>column</u>) is constructed from boxes of two sizes.

A <u>type 1 row</u> (<u>column</u>) is one in which each box has a dimension which fills the height (width) of the row (column).

A <u>type 2 row</u> (<u>column</u>) is one in which at least one portion of the row (column) is completed by combining boxes to fill the height (width) of the row (column) within tolerance.

A <u>modular cluster</u>, or <u>module</u>, is a cluster obtained by combining partial rows and columns.

A <u>symmetric modular cluster</u> is a modular cluster having a symmetric arrangement of rows and columns, an exterior rectangular boundary, and an interior space, if any, which is rectangular.

A <u>repeated symmetric modular cluster</u>, or <u>repeated cluster</u>, is a cluster obtained by repetition of a symmetric modular cluster.

An <u>asymmetric modular cluster</u> is a modular cluster having a non-symmetric arrangement of rows and columns, no interior space, and an exterior boundary which is rectangular except for the possible presence of a corner rectangular space.

A <u>cluster seed</u> is a cluster which is considered to have the highest value of all clusters to be combined in the formation of a particular configuration.

Pattern formation progresses in an orderly sequence of steps. Although patterns are not fixed with respect to box dimensions, the types of configurations to be included in a pattern are fixed. Each pattern is constructed by beginning with box seeds and developing clusters. Cluster seeds are then combined with other clusters, according to predetermined rules, to form specific types of configurations. Patterns of simpler configuration are sought first, and if they cannot be used, more elaborate types are considered. Once again, specific rules will be enforced to limit the amount of work expended in seeking the desired result and also to eliminate unwarranted iterations. Pattern formation for all boxes in the loading population is not guaranteed. When failure occurs, the task will be returned to the analyst for further consideration.

Before even simple patterns are formed, complete rows and columns must be generated. The procedure for generating complete rows relies extensively on the integer linear combination technique already discussed.

<u>Row Completion Procedure</u>

Step 1: Select as the seed box the next remaining box of maximum value; let its side dimensions be a and b where $a \geq b$.

Step 2: Try to generate uniform complete rows.

a. Attempt to form a type 1 uniform complete row of height a, using an integer linear combination of b. If unsuccessful,

and a and b have not been interchanged, then interchange them and repeat Step 2a; otherwise restore the original values of a and b and go to Step 2b.

b. Attempt to form a type 2 uniform complete row of height a by first obtaining an integer linear combination of b to generate columns within tolerance of height a and, if successful, taking integer linear combinations of a and b to complete the row. If unsuccessful, and a and b have not been interchanged in Step 2b, then interchange them and repeat Step 2b; otherwise restore the original values of a and b and go to Step 3.

Step 3: Try to generate non-uniform complete rows.

a. Attempt to form a type 1 non-uniform complete row of height a using an integer linear combination of dimensions b and d of two different boxes; dimension b is that of the seed box having known lateral dimensions a x b and d is unknown and belongs to a box having lateral dimensions a x d where $d < b$. If unsuccessful, and a and b have not been interchanged, interchange them and repeat Step 3a; otherwise restore the original values of a and b and go to Step 3b.

b. Attempt to form a type 2 non-uniform complete row of height a by first obtaining an integer linear combination of c, where c is unknown and c < a, to generate columns within tolerance of height a and, if successful, taking integer linear combinations of two dimensions; dimension b is that of the seed box having dimensions a x b and dimension d is that of a box having dimensions c x d for which columns of height a have been successfully generated. If unsuccessful, and a and b have not been interchanged in Step 3b, then interchange them and repeat Step 3b; otherwise, restore the original values of a and b and go to Step 4.

Step 4: Flag failure; a complete row cannot be constructed for box seed of size a x b.

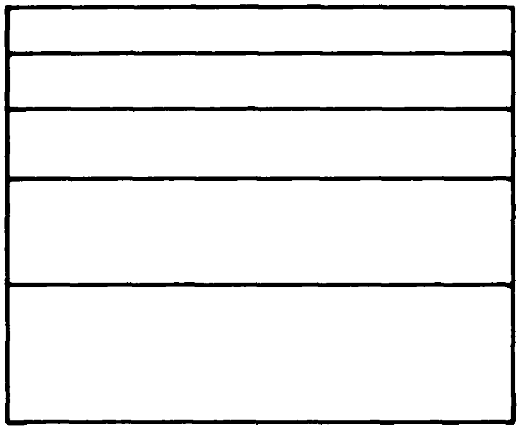
The generation of complete columns is analagous to that for complete rows with row and column dimensions reversed. A list of complete rows and a separate list of complete columns is generated with its members arranged in decreasing sequence of row heights and column widths.

Patterns are characterized as simple or complex, depending on the types of configurations from which they are constructed. (Configuration types differ from the previously defined row types.) Patterns synthesized

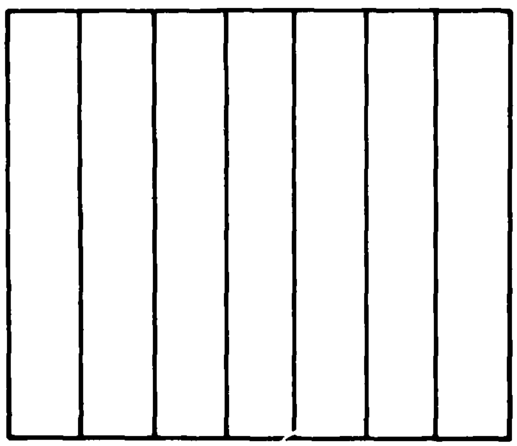
from a type 1 or type 2 configuration are simple; patterns derived from type 3, type 4, or combinations of these types are considered to be complex.

A type 1 configuration is developed by integer linear combination of either all complete rows or all complete columns. If the configuration has only complete rows, it is called type 1A, and if it consists of only complete columns, it is called type 1B. (If a configuration is identified as type 1A, it will be so designated and it will be immaterial whether that configuration simultaneously satisfies the criteria for type 1B.) The type 1 configuration is the first to be sought. Examples of a type 1 configuration are given in Figure 4.

The procedure for generating a type 1 configuration is almost identical to that previously described for the row completion procedure and that inferred for the column completion procedure. A type 1A configuration can be looked upon as a complete column derived by integer linear combination of boxes having width equal to the complete row width. The seed is the row having the largest row height. A type 1B configuration can be considered as an integer linear combination of boxes having length equal to the complete column height. The seed is the column having the largest column width.



TYPE 1A Configuration obtained by LINEAR combination of complete rows



TYPE 1B Configuration obtained by LINEAR combination of complete columns

Figure 4 - Type 1 Configuration

Type 1 Configuration Formation Procedure

Step 1: Attempt to exhaust the list of complete rows by formation of type 1A configurations using integer linear combinations.

Step 2: Attempt to exhaust the list of complete columns by formation of type 1B configurations using integer linear combinations.

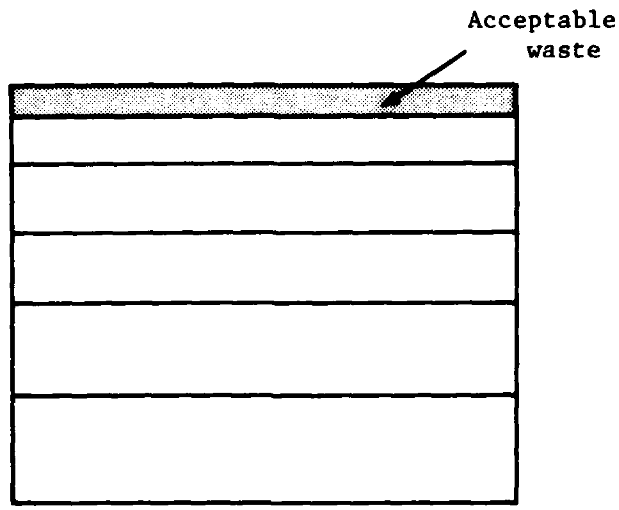
Step 3: Make any remaining complete rows or columns available for subsequent type 2 configuration analysis.

A type 2 configuration is developed by either complete rows or complete columns followed by refinements for improving space utilization. A type 2A configuration is derived from complete columns. Examples of the type 2 configuration are given in Figure 5. The formation of the type 2 configuration places greater value on complete row and complete column clusters than on boxes which are not clustered. This priority of values helps to further reduce the number of unused complete rows and columns remaining after the type 1 configurations have been formed.

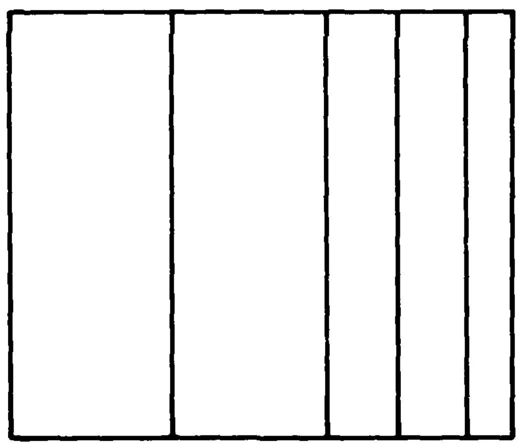
Type 2 Configuration Formation Procedure

Step 1: Use the list of unused complete rows and select in succession as many rows as possible without overfilling the available space.

Step 2: Try to reduce the unused pattern space by trading the next available unused complete row for any previously selected row, other than the seed row, having lesser height.



Type 2A configuration obtained by
sequential access of complete row list



Type 2B configuration obtained by sequential
access of complete column list

Figure 5 - Type 2 Configuration

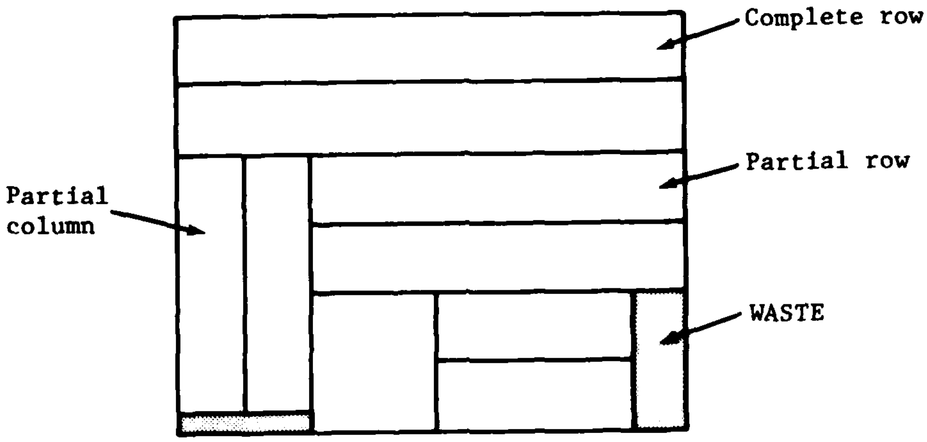
Step 3: Repeat Step 2 twice if necessary. At this point, if the cluster seed has not generated an acceptable type 2A configuration, this type of configuration need no longer be pursued with the same seed.

Step 4: Repeat Steps 1 through 3 to obtain as many type 2A configurations as possible.

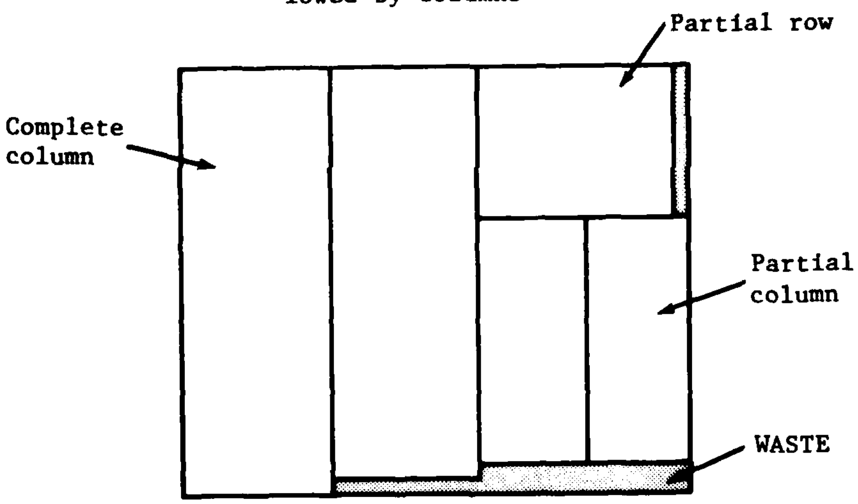
Step 5: Repeat Steps 1 through 4 using complete columns instead of complete rows to obtain as many type 2B configurations as possible.

The type 2 configuration formation procedure is followed by the type 3 configuration formation procedure, which attempts to synthesize an asymmetric modular cluster. The type 3A configuration is generated by exhausting the list of unused rows and alternately filling the remaining space with partial columns and rows. This type of configuration is illustrated in Figure 6.

In the previous type 1 and type 2 configuration formation procedures, a configuration was considered successful when it filled the available pallet space within tolerance. In that case, the configuration produced a complete pattern; for both types 1 and 2, it was advantageous to repeat the procedures to obtain as many complete patterns as possible.



Type 3A configuration
obtained by alternate
filling of rows fol-
lowed by columns



Type 3B configuration
obtained by alternate
filling of columns fol-
lowed by rows

Figure 6 - Type 3 Configuration

In the type 3 configuration formation procedure success does not necessarily imply full use of the available space. Rather, a promising configuration is further analyzed and possibly combined with other configurations to synthesize a pattern which produces efficient utilization of the available space. If a type 3 configuration is not within tolerance, the major portion of the wasted space will always be in one corner of the pattern. That corner space should be filled with a type 4 configuration before returning to a type 3 configuration procedure for further improvement or for beginning a new pattern. Type 3A and type 3B configurations are generated in an identical manner except that the type 3A formation begins by considering unused column space, and the type 3B first considers unused row space.

Type 3 Configuration Formation Procedure

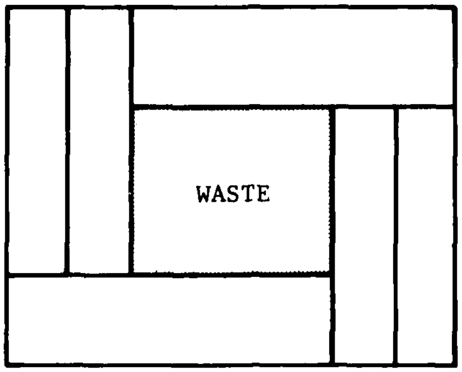
Step 1: Exhaust any remaining complete rows (columns). If none, go to type 4 procedure; otherwise choose next available box seed that fits into remaining available space and go to Step 2 (Step 3).

Step 2: Try to form as many partial columns as possible to reduce waste width. If successful (that is, at least one partial column is found) and both corner waste and pattern waste are within tolerance, stop. If successful and corner waste is out of tolerance, go to Step 3; otherwise, go to Step 4.

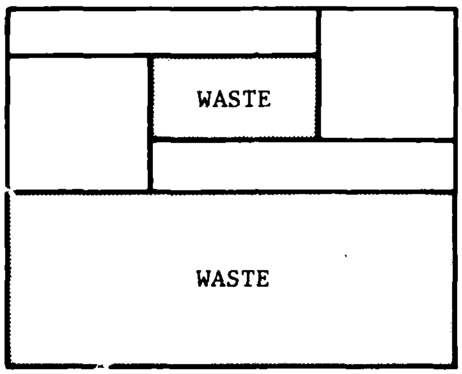
Step 3: Try to form as many partial rows as possible to reduce waste height. If successful (that is, at least one partial row is found) and both corner waste and pattern waste are within tolerance, stop. If successful and corner waste is out of tolerance, go to Step 2; otherwise go to Step 4.

Step 4: If at least one success is achieved for type 3 (in either Step 2 or Step 3), flag the next box seed for type 4 as variable. If no success is achieved for type 3, flag the seed selected in Step 1 for retention; attempt further completion of the pattern by going to the type 4 configuration procedure.

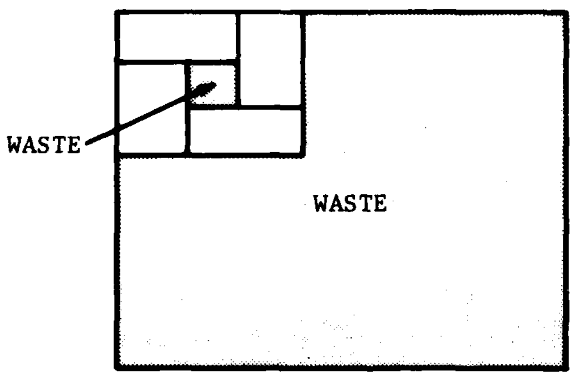
The type 4 configuration, the most intricate type of configuration to be sought, is characterized by the presence of a symmetric modular cluster. There are three subtypes of this configuration. The first is formed by a symmetric integer linear combination of two sizes of boxes. An example of this subtype, type 4A, is given in Figure 7A. This type of combination represents two distinct linear combinations, one for each of the two dimensions of the available space. In this type of combination, both the height and width of the available space must be decreased and any resultant waste space must be confined to the interior of the original available space. If the resulting waste space is not within tolerance, then an attempt is made to further reduce the waste by returning to the type 3 configuration procedure and trying to insert a type 3 configuration



Type 4A configuration fills outer portion of space in 2 dimensions



Type 4B configuration fills available space in only 1 dimension



Type 4C configuration does not fill available space in either dimension

Figure 7 - Type 4 Configuration

into the wasted space. An example of such nesting of type 3 and type 4 configurations is given in Figure 8.

The second subtype is similar to the first subtype except that, as a result of the symmetric integer linear combination, a symmetric modular cluster is obtained which reduces only one of the dimensions of the original available space. The second dimension is reduced by repetition, if possible, of this modular cluster. An example of this second subtype, type 4B, is shown without repetition in Figure 7B. The resultant waste space consists of an interior hole and a remainder wasted space. Filling of these spaces is attempted, once again, by a type 3 configuration.

The third subtype, a further generalization in configuration formation, is formed as a result of symmetric integer combination of two box sizes which does not fill either the height or width of the original available space. This type of configuration includes a further restriction that the resultant interior waste space must be so small that no additional interior fill is needed. Repetition of the modular cluster formed by this combination is sought in two dimensions to reduce the available space. Two rectangular remainder spaces may result, each parallel to an edge of the original space. An example of this subtype, type 4C without repetition, is given in Figure 7C.

<u>Type 4 Configuration Formation Procedure</u>

Step 1: Choose next suitable box seed.

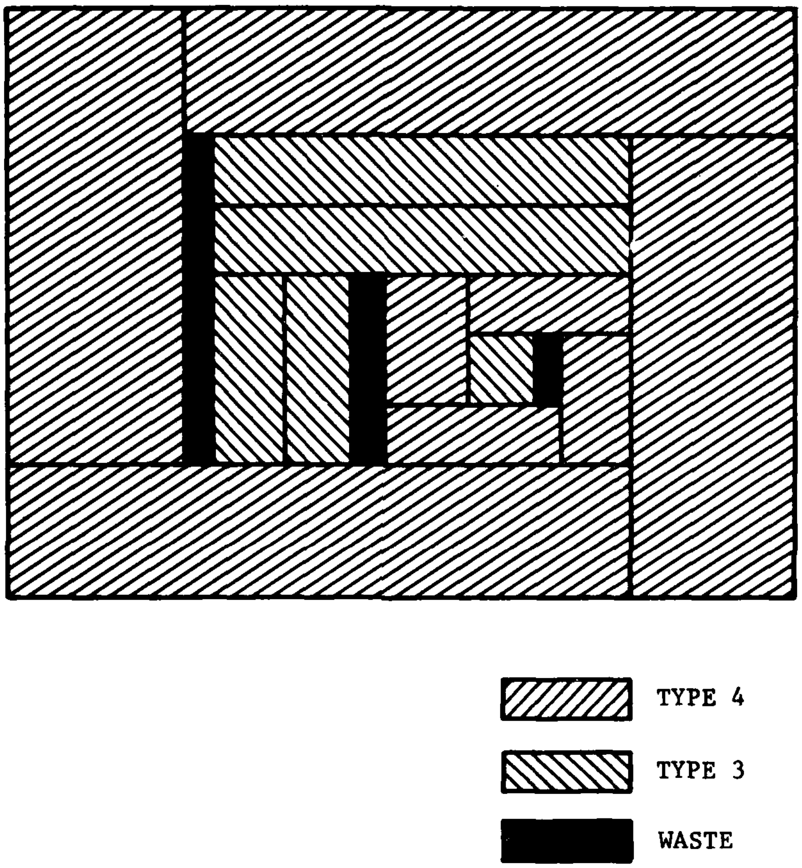


Figure 8 - Pattern Completion by Nesting of Type 3 and Type 4 Configurations

Step 2: Try to obtain type 4A configuration by a symmetric integer linear combination of two box sizes which is as large as both the height and width of the available space. If successful, and within tolerance, stop. If successful, and out of tolerance, go to type 3 procedure (omitting Step 1) to further reduce waste; otherwise, go to Step 3.

Step 3: Try to obtain type 4B configuration by a symmetric integer linear combination of two box sizes which is as large as either the height or width of the available space. If successful, repeat as often as possible in the other direction. If successful and within tolerance, stop. If successful and out of tolerance, go to type 3 procedure (omitting Step 1) to further reduce waste; otherwise go to Step 4.

Step 4: Try to obtain type 4C configuration by a symmetric integer linear combination of two box sizes with a small interior hole. If successful, repeat as often as possible, to fill both the height and width of the available space. If successful and within tolerance, stop. If successful and out of tolerance, go to type 3 procedure (omitting Step 1) to further reduce waste; otherwise flag failure and stop.

The four types of configurations which have been described may be used to generate countless numbers of patterns. The actual patterns and

the specific box sizes will be governed by the available load population. All of the patterns in Appendix B can be generated. The sequence of steps in generating the various configurations gives priority to the formation of patterns which are uniform, that is, all boxes in a pattern will be of one size if sufficient numbers are available in the load population.

When a promising type 3 or type 4 configuration cannot be completed and the cluster seed cannot be efficiently loaded, then the analyst will be given the opportunity to intercede and possibly to improve the incomplete pattern. The same holds for any pattern which is incomplete and cannot be completed because the load population is exhausted.

Stacking

The stacking stage determines the sequence in which tiers of boxes, loaded in accordance with the previously formed patterns, will be stacked. (Refer to Figure 2, Step 4 for stacking terminology.) The height of the space available for stacking is the interior height of the container. The stacking sequence is subject to the following additional constraints:

- height limitation of the load on an individual pallet
- allowance for heights of pallets to be inserted into stack
- allowance for necessary waste to permit accessibility

- weight limitation of the load on an individual pallet

- stability of the stack

The stacking procedure initially ignores the weight and stability constraints and adjusts for them later. Since it is advantageous to keep the number of items involved at one time as small as possible, the procedure used will develop subsequences for preliminary formation of individual pallet piles, preferably to their maximum permissible heights. These subsequences are then combined and modified to obtain a sequence that spans the total effective height. In this way, a full stack is obtained more quickly by stacking piles instead of individual tiers.

Even though the objective is to produce full preliminary pallet piles, the initial procedure is done rapidly with no attempt at refinements. Selected piles are stacked one upon the other and, when necessary, additional tiers not included in any piles are added to the stacks. The procedure emphasizes the formation of full stacks since, once they are obtained, it is immaterial whether two partially loaded pallets or a fully loaded pallet and a partially loaded pallet are used if in

both instances the two pallets spanned the same stack height, as shown in Figure 9.

The following definitions will be needed in the discussion of the stacking procedure:

HSTACK = stack height of load including pallets

HTFMAX = maximum permissible height of a full pile including pallet height

HTPALL = pallet height

HTCONT = effective height of container (interior height of container - waste height required for accessibility)

TOLPIL = allowable tolerance on a full pile (minimum height of a full pile = HTFMAX - TOLPIL.)

TOLSTK = allowable tolerance on a full stack

The minimum number of pallets, NMIN, required to span the effective container height is, therefore:

$$\text{NMIN} = \left| \frac{\text{HTCONT}}{\text{HTFMAX}} \right|$$

where $|\ \ |$ denotes next largest integer.

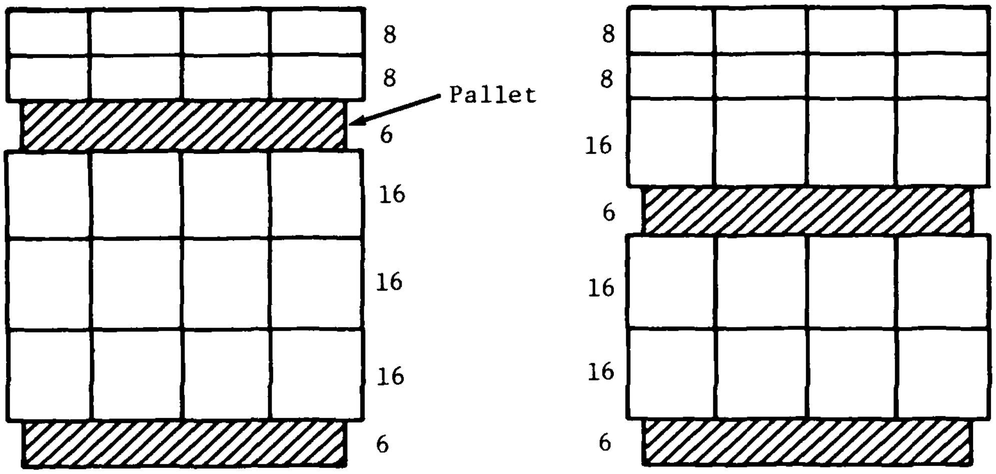
The easiest way to build a full stack with the minimum number of pallets is to stack NMIN - 1 full piles and a partially loaded pallet with a minimum height, HTPMIN, given by:

$$\text{HTPMIN} = \text{HTCONT} - (\text{NMIN} - 1) \times (\text{HTFMAX} - \text{TOLPIL})$$

provided

$$\text{HTPMIN} < \text{HTFMAX} - \text{TOLPIL}.$$

(If HTPMIN > HTFMAX - TOLPIL, simply stack NMIN full piles.)



60" Stack Containing
1 full pallet load
1 partial pallet load

60" Stack containing
2 partial pallet loads

Figure 9 - Alternative Pallet Loads Spanning Identical Stack Height

As many full piles as possible are formed from tiers contained in a list of available tiers which is arranged by decreasing tier heights. Successive tiers are selected until either the list of available tiers is exhausted or the addition of the next tier will cause the pile to exceed the maximum permissible pallet height, HTFMAX. If the pile height, HTPILE, is sufficiently high to be considered full, i.e., if

$$\text{HTFMAX} - \text{HTPALL} - \text{TOLPIL} < \text{HTPILE} \leq \text{HTFMAX} - \text{HTPALL}$$

then the sequence of tiers constituting the full pile is saved by putting it on a list of full piles and updating the total number of full piles, NFULL. If the pile is not full, then a test is made to determine whether

$$\text{HTPMIN} - \text{HTPALL} \leq \text{HTPILE} \leq \text{HTFMAX} - \text{HTPALL} - \text{TOLPIL}$$

If so, the sequence of tiers is saved on a list of partial piles, and the total number of partial piles, NPART, is updated. If, however, HTPILE < HTPMIN - HTPALL, then the tiers are placed on a special list of unused tiers having NUNUSE entrees. Thus, the desired lists of full piles, partial piles, and unused tiers are generated without excessive expenditure of time and money in rearranging tiers.

The process for stacking piles and unused tiers is described in terms of system states which are defined on the basis of the number of entries in each of these lists. The system states are defined by the system conditions given in Table 1. The state transition matrix is given in Table 2. A check mark (✓) denotes that a transfer is possible between

TABLE 1 - SYSTEM STATE DEFINITIONS

State	System Condition
1	NPART > 0, NFULL > NMIN-1
2	NPART = 0, NFULL > NMIN
3	NPART > 0, 0 < NFULL ≤ NMIN
4	NPART = 0, NFULL ≤ MIN
5	NFULL = 0, NPART > NMIN
6	NFULL = 0, 0 < NPART ≤ NMIN
7	NFULL = 0, NPART = 0, NUNUSE

TABLE 2 - STATE TRANSITION MATRIX*

FROM \ TO	1	2	3	4	5	6	7
1	✓	✓	✓	✓	✓	✓	✓
2	X	✓	X	✓	X	X	✓
3	X	X	X	✓	✓	✓	✓
4	X	X	X	X	X	X	✓
5	X	X	X	X	✓	✓	✓
6	X	X	X	X	X	X	✓
7	X	X	X	X	X	X	✓

*✓ denotes state transition can occur
X denotes state transition cannot occur

the "from" and "to" states, and an "X" denotes that a transfer is not possible. The significance of states, state transitions, and their effect on the stacking procedure is explained as follows:

Under the system conditions for state 1, that is, NPART > 0, NFULL > NMIN - 1, the stack is built by stacking, in succession, the next available NMIN - 1 full piles and placing uppermost a single partially loaded pile. At this point, because of the manner in which full and partial piles were defined and generated, the stack must either be full

$$\text{HTCONT} - \text{TOLSTK} < \text{HSTACK} \leq \text{HTCONT}$$

or overfilled

$$\text{HTSTACK} > \text{HTCONT}.$$

The height of the stack, HSTACK, is given by

$$\text{HSTACK} = \text{HTPART}(1) + \sum_{I=1}^{\text{NMTN}-1} \text{HTFULL}(I) + \text{NMIN X HTPALL}$$

where the first term on the right is the height of the partial pile, the second term represents the sum of the heights of the full piles, and the last term represents the space to be occupied by pallets for supporting the piles, provided that the stack contains the minimum number of pallets. At this stage of the loading process, it is convenient to set aside space for NMIN pallets, but it must be emphasized that the hypothetical stack consists only of tiers which have been tentatively grouped into piles without their supporting pallets. The overfill height, HOVER, is given by

$$HOVER = HSTACK - HTCONT$$

In state 1, HOVER $\geq$ 0 and an unloading procedure is needed to remove one or more tiers to make the stack height as close as possible to, but still below, the permissible stack height. Tier removal starts at the uppermost pile and, if an underfill condition within tolerance is not achieved, lower piles are tested until success is attained. The Nth pile is unloaded by removing the largest possible tier which will either create an underfill with the least waste, or, if an underfill is impossible, by removing the largest possible tier to create a stack with the least overfill.

Mathematically, for all I tiers in pile N with tier heights HTTIER (N,I), compute

$$HTTIER(N,I) - HOV(N)$$

where

$$HOV(N) = HOVER$$

and select pattern J such that, for all I,

$$0 \leq HTTIER(N,J) - HOV(N) \leq HTTIER(N,I) - HOV(N).$$

If J is void, then an underfill cannot be obtained by removing just one tier and, therefore, a Kth tier must be temporarily removed so that:

$$HTTIER(N,I) - HOV(N) \leq HTTIER(N,K) - HOV(N) < 0$$

Once again tier J is sought to obtain the desired underfill. When the Jth tier is found, test whether

$$HTTIER(N,J) - HOV(N) < TOLSTK.$$

If so, permanently remove the patterns selected for the desired underfill. If the patterns to be removed do not result in an acceptable underfill, then proceed to analyze the next lower pile. If all piles in the stack have been analyzed without finding one that fills the requirements, then use the best possible pile for unloading, even if the result is not within specified tolerance.

The type of logic that has been used is typical of the stacking procedure for all states. The specific procedure depends on the number of entries in each of the lists and the sequence in which the lists become exhausted. If, for example, the state 1 stacking procedure is repetitively applied and results in the exhaustion of the list of partial piles, that is, NPART = 0, then a transition of the system state would occur, and either state 2, 4, or 7 could result.

When all lists are available, the procedure gives priority to using, simultaneously, full piles and partial piles (state 1). If no more full piles are available, the procedure uses as many partial piles as possible before modifying the stack with individual tiers selected from the list of unused tiers. If the partial piles are exhausted before the full piles, then preference is given to using as many full piles as possible before accessing the list of unused tiers. In the absence of any full or partial piles (state 7) the complete stack is built with individual tiers.

In state 1, an overfull condition results most frequently, but in other states this is not necessarily the case. In state 7, for instance, individual tiers are selected and a height computation is made to ensure that an overfill condition never results. In the other states, either an overfill or underfill condition may initially result. If an underfill results, then a space filling procedure is required which consists of using the list of unused tiers and adding as many successive tiers as possible until the addition of the next tier causes an overfill. In other words, the space filling procedure reduces wasted space by approaching the acceptable full stack height from below and never exceeding the desired result. For all states, the common goal is to fit the stack into the available space and then reduce waste in a stepwise procedure which always results in improvement.

An additional refinement is available which, if it can be used, will always result in improvement. If a stack is underfilled and not within tolerance, and the space filling procedure has failed to put the stack within tolerance, the next available tier on the list of unused tiers is substituted for an existing tier in the stack. If the height of the next available tier is HNEW, and the height of an existing tier in the stack is denoted as H(I), then any existing tier in the stack which is a candidate for trading must satisfy the criterion

$$0 < \mathrm{HNEW} - \mathrm{H(I)} \leq \mathrm{HWASTE}$$

where

$$HWASTE = - HOVER$$

The most desirable existing tier to be traded, J, satisfies the criterion that

$$0 < HNEW - H(I) \leq HNEW - H(J) \leq HWASTE.$$

Additional details on stacking are given in the flowcharts of Appendix C. In certain states the list of unused tiers may be exhausted or nearly exhausted but individual tiers are needed to complete the stack under analysis. Under these circumstances, an available partial pile may be disassembled and its constituent patterns placed on the list of unused tiers. If no partial pile is available, an available full pile can be disassembled and handled in like manner. If a stack is too low and neither full piles, partial piles, or unused tiers are available, the entire load has been stacked.

Completion

The first step in the completion stage is the insertion of pallets into the previously formed stacks. This task is simplified by the procedure used to form a stack. If the stack contains any full piles, they will be at the bottom of the stack, and pallets are first inserted beneath each full pile.

If a stack contains both partial piles and individual tiers not assigned to any pile, the procedure could be to consolidate partial piles

and add tiers to the remaining partial piles, or to add tiers to partial piles and then consolidate. Since both methods have disadvantages, a procedure was developed which represents a mixture of these methods. If the number of full piles plus partial piles exceeds the minimum number of pallets required to load a full stack, partial piles are consolidated before any remaining tiers are loaded. After this initial consolidation, any additional tiers are added to these partial piles and, if all partial piles are exhausted, a new pile is started from the remaining tiers on the list of unused tiers. At no time is a partial pile permitted to become overstacked. When this list is exhausted, all piles except the original full piles are consolidated and a pallet is inserted beneath each remaining pile. In the rare instance in which the number of pallets exceeds the minimum that could be used and, simultaneously, the resulting stack height after pallet insertion is excessive, the most convenient tier having a height which exceeds the overstack is removed and placed on the lowest remaining stack. Each stack is then analyzed by the same procedure until pallets have been appropriately placed into all stacks.

The weight of each pallet and its stability are considered next. The computer is ideally suited for monitoring the total weight of all boxes on a pallet. If a pallet's weight is excessive, the situation is remedied by removing part of its load, beginning at the topmost tier of boxes. Any boxes so removed are rescheduled for loading on another

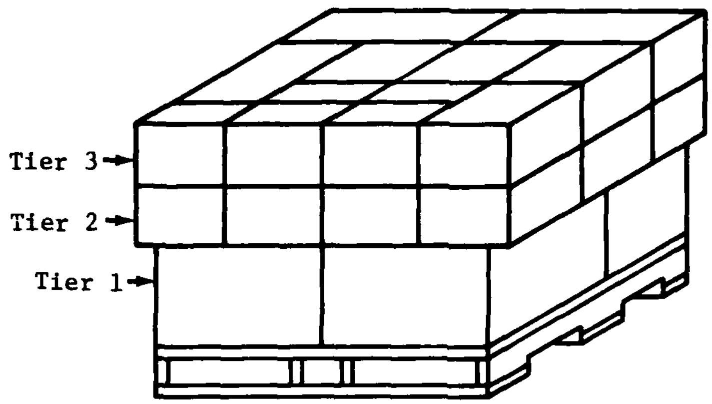
pallet. To increase the stability of a pallet, tiers placed on a pallet are rearranged in decreasing order of size starting at the bottom of a pallet. This ensures that upper tiers do not excessively overlap lower tiers and also facilitates pallet load strapping and prevents load shifting. In stacking tiers formed repetitively from the same pattern, the rotation of successive tiers further stabilizes a pallet load. Figure 10 shows the improvement of pallet load stability; tier 3, with the same pattern as tier 2, is rotated 180 degrees and tier 1, with a smaller area than the other two tiers, is placed on top.

Another program feature for enhancing load stability will provide information on necessary interior supports to be placed in interior spaces of tiers derived from patterns having interior wasted spaces; this prevents load shifting.

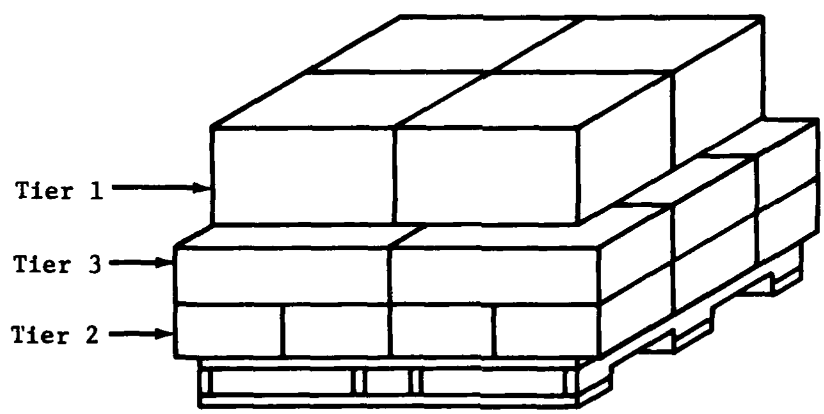
The final task of the completion stage generates an output report which describes in detail all patterns, tier sequencing on each pallet, and the space utilization measures of each pallet and of each stack.

CONTAINER LOADING

After completion of pallet loading, an analytic procedure will provide an efficient spatial arrangement of the stacks for each size container, monitor the total weight of the total load in each container, and provide for load stability. The tasks associated with this procedure are collectively termed "container loading."



Original pallet load



Stabilized pallet load
obtained by rotating Tier 3
and placing Tier 1 uppermost

Figure 10 - Improvement of Pallet Load Stability

All the essential features of the container loading procedure become available as by-products of the previously developed pallet loading procedure. If all pallets are the same size and if a stack is temporarily considered as one tall box, then determining the arrangement of stacks is really the same problem as finding a pattern for arranging boxes of identical size. For this purpose, the type 1 configuration formation procedure previously described is most suitable. For the present application, however, the procedure is more restrictive, since, if only one size pallet is used, only uniform rows and columns are generated, and the resulting configuration is considerably simpler.

The total weight of the load in each container is obtained by simply summing the weights of the stacks. The stability of a container can be improved by interchanging stacks to offset imbalance of the load resulting, for example, when heavier stacks predominate on one side of a container. A report of the container load describing in detail the arrangement of stacks and the space utilization measure of each container will also be generated.

BARGE LOADING

The procedures for arrangement of containers on a barge and for checking weight and stability are virtually identical to the container loading procedure. If containers of more than one size are to be loaded, the type 1 pattern to be used for their arrangement need not necessarily

be derived from uniform rows and columns. Each rectangular space on the barge available for loading will be analyzed independently. A final report will be generated denoting the arrangement of containers, the total weight, and the overall space utilization efficiency measured with respect to the total available space on the barge.

ORGANIZATION AND FUNCTIONAL DESCRIPTION OF PROPOSED PROGRAM LOADER

The organization of main program LOADER, its subroutines, and their interrelationships is provided in Appendix C. A flowchart of the main program is given in Figure 13, and flowcharts for most of the subroutines appear in Figures 14 - 32. A brief description of the function of each routine follows.

LOADER	-	main program; performs executive role of controlling, sequencing, and monitoring program operations.
INPUT	-	reads input data and generates input data report.
ANALYZ	-	analyzes input data and rearranges data into most convenient form.
PATRNS	-	forms patterns by generating and combining clusters; assigns patterns to tiers.
STRING	-	generates complete row and complete column clusters.
TYPE 1	-	generates type 1 configurations.
TYPE 2	-	generates type 2 configurations.
TYPE 3	-	generates type 3 configurations.
TYPE 4	-	generates type 4 configurations.
SAVE	-	creates a file containing data describing complete patterns.
STACKS	-	controls formation of stacks generated by stacking full piles, partial piles, and unused tiers.
PILES	-	generates preliminary full piles and partial piles.
STATE	-	determines the system loading state from the available number of preliminary full piles, partial piles, and unused tiers.

STAKER - forms stacks from full piles, partial piles, and unused tiers; stacking strategy used depends upon system loading state.

SUPPLY - checks and makes available, if possible, a specified minimum number of unused tiers.

FILL - fills available wasted space in a stack by rapid selection of unused tiers.

UNLOD - removes tiers from stacks which are overloaded.

TRADE - attempts refinement of available wasted space in a stack by trading tiers.

PALETS - forms complete pallet loads by consolidating piles, inserting pallets beneath piles, and making final stack adjustments, if necessary.

WEIGHT - monitors weight constraints in stacks, containers, and barges and assists in corrective action.

STABLE - monitors stability in stacks, containers, and barges and assists in corrective action.

CNTNER - controls container loading operations.

BARGE - controls barge loading operations.

REPRT1 - generates report of composition of all stacks including the arrangement of all boxes in each tier.

REPRT2 - generates report on arrangement of stacks in each container.

REPRT3 - generates report on arrangement of containers in barge.

SAMPLE PROBLEM

STATEMENT OF PROBLEM

Containers #1 and #2 are partially loaded with cargo for destinations A and B, respectively. Cargo for container #1 is restricted to a maximum loaded pallet height of 48 inches and for container #2 to 36 inches. Additional cargo for destination C is to be loaded completely on either container #1 or #2. The following additional load data apply:

Pallet dimensions: 48" W X 40" L

Permissible pallet dimensions with overhang: 52" W X 43" L

Pallet height: 6"

Available space in container #1:

Effective height	- 90"
Width	- 90"
Length	- 54"

Available space in container #2:

Effective height	- 84"
Width	- 90"
Length	- 54"

Characteristics of cargo for destination C:

Box Dimensions (Inches) L X W X H	Quantity	Box Dimensions (Inches) L X W X H	Quantity
28 X 16 X 16	2	24 X 21 X 15	2
42 X 12 X 14	1	13-1/2 X 9 X 12	8
37 X 12 X 14	2	18 X 6 X 8	4
36 X 14 X 8	2	15 X 15 X 10	4
45 X 38 X 10	1	11 X 11 X 11	8
28 X 6 X 12	4	11 X 7-1/2 X 15-1/2	8
28 X 22 X 14	2	8 X 6 X 16	6
26 X 14 X 16	1	8 X 6 X 10	6
26 X 12 X 16	3	15 X 15 X 14	4
22 X 11 X 10	4	14 X 12 X 18	12
22 X 15 X 12	4	18 X 14 X 14	2
20 X 14 X 8	2	6 X 6 X 6	8

The cost of shipment by container #2 is less than for #1 and it is therefore preferable, if possible, to use container #2. The problem is to determine analytically whether the cargo can be loaded in each of the available spaces, to compare their space utilization, and to examine the effect of the more restrictive maximum loaded pallet height in container #2.

SOLUTION

Assumptions and Restrictions

1. Boxes must be loaded with height dimensions oriented vertically.
2. Group height tolerance for box height, h:

Height Range	Tolerance (inches)
$h > 40$	5
$30 < h \leq 40$	4
$20 < h \leq 30$	3
$10 < h \leq 20$	2
$h < 10$	1

3. Cluster waste tolerances:

    10% for all row and column clusters

    10% for holes in symmetric modular clusters

    10% for corner waste in asymmetric modular clusters

    15% for outerfill waste of symmetric modular clusters

4. Pattern waste tolerances:

    10% for simple patterns (types 1 and 2)

15% for complex patterns (types 3, 4, or combinations)

Stack tolerances:

3" for stack height waste

6" for preliminary full piles

Preliminary Load Data Analysis

The maximum box height is 18" and the minimum is 6". *Define groups* by range of box heights, h:

Group	Height Range (Inches)
1	$16 < h \leq 18$
2	$14 < h \leq 16$
3	$12 < h \leq 14$
4	$10 < h \leq 12$
5	$9 < h \leq 10$
6	$8 < h \leq 9$
7	$7 < h \leq 8$
8	$6 < h \leq 7$

The load population consists of 100 boxes which will be analyzed collectively, that is, batch size = 100. The following table is obtained by grouping the batch data, computing the box areas and volumes, sequencing the groups by decreasing group heights, and sequencing the boxes within each group by decreasing areas.

TABLE 3 - SAMPLE PROBLEM LOAD POPULATION DATA

Group	Group Height* (inches)	Box Dimensions (inches) L X W X H	Box Area (square inches) L X W	Box Volume (cubic inches) L X W X H	Quantity
1	18	14 X 12 X 18	168	3024	12
2	16	24 X 21 X 15	504	7560	2
		28 X 16 X 16	448	7168	2
		26 X 14 X 16	364	5824	1
		26 X 12 X 16	312	4992	3
		11 X 7-1/2 X 15-1/2	82.5	1279	8
		8 X 6 X 16	48	768	6
3	14	28 X 22 X 14	616	8624	2
		42 X 12 X 14	504	7056	1
		37 X 12 X 14	444	6216	2
		18 X 14 X 14	252	3528	2
		15 X 15 X 14	225	3150	4
4	12	22 X 15 X 12	330	3960	4
		28 X 6 X 12	168	2016	4
		13-1/2 X 9 X 12	121.5	1458	8
		11 X 11 X 11	121	1331	8

* Group Height is maximum box height of all boxes in a group.

TABLE 3 (Continued)

Group	Group Height* (inches)	Box Dimensions (inches) L X W X H	Box Area (square inches) L X W	Box Volume (cubic inches) L X W X H	Quantity
5	10	45 X 38 X 10	1710	17100	1
		22 X 11 X 10	242	2420	4
		15 X 15 X 10	225	2250	4
		8 X 6 X 10	48	480	6
6					NONE
7	8	36 X 14 X 8	504	4032	2
		20 X 14 X 8	280	2240	2
		18 X 6 X 8	108	864	4
8	6	6 X 6 X 6	36	216	8

* Group Height is maximum box height of all boxes in a group.

Pallet Loading

Each group is processed independently to form patterns; intermediate diagrams are provided for instructional purposes but are unnecessary for carrying out any of the procedures.

Group 1 Processing

Group 1 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
18	14 X 12	12

Group 1 contains only one box size, and the box seed is, therefore, 14 X 12. Apply row completion procedure.

Step 1: $a = 14$, $b = 12$

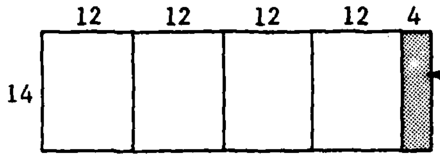
Step 2: Generate uniform complete rows.

Try to form a type 1 uniform complete row of height 14 using an integer linear combination of 12. First iteration of integer linear combination technique (problem version A in section entitled "Proposed Approach") applied by considering $X_1 = b = 12$ and x_2 unknown, $n_{1AVAIL} = 12$, gives $n_1 = \min \frac{52}{12}, 12 = 4$, $n_2 = 0$. Since complete row tolerance is 10% and maximum permitted pallet load width = 52, TOL = .10(52) = 5.2. Substituting in

$$X - TOL \leq n_1 x_1 \leq X$$

$$52 - 5.2 \leq 4(12) \leq 52$$

This statement is valid, $n_1 = 4$ and $n_2 = 0$ are optimum, and a successful type 1 uniform complete row is obtained.

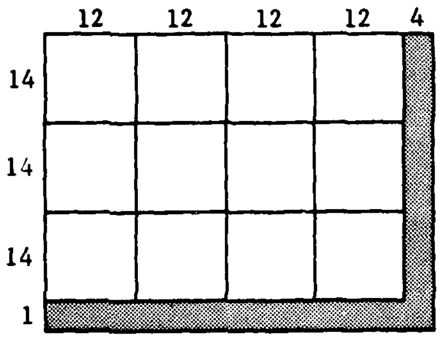


$$\text{Row waste} = \frac{\text{row waste area}}{\text{available row area}} = \frac{4 \text{ X } 14}{52 \text{ X } 14} \text{ X } 100 = 7.6\%.$$

Since there are eight more boxes of the same size, two additional complete rows of the same configuration may be immediately generated. There are no remaining boxes in group 1, so proceed to pattern formation and begin with type 1 configuration.

Step 1:

All complete rows have the same row height, so the cluster seed has height = 14. Try to form a type 1A configuration by integer linear combination (problem version A) of $x_1 = 14$, x_2 unknown, $n_{1AVAIL} = 3$, which immediately gives $n_1 = 3$, $n_2 = 0$ and the following acceptable complete pattern is obtained:



PATTERN NO. 1

$$\text{pattern efficiency} = \frac{\text{actual load area}}{\text{maximum pallet load area}} = \frac{12 \text{ X } 168}{43 \text{ X } 52} \text{ X } 100 = 90.2\%$$

Pattern waste = 100 - 90.2 = 9.8%.

Group 1 is exhausted; go to next group.

Group 2 Processing

Group 2 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
16	24 X 21	2
	28 X 16	2
	26 X 14	1
	26 X 12	3
	11 X 7-1/2	8
	8 X 6	6

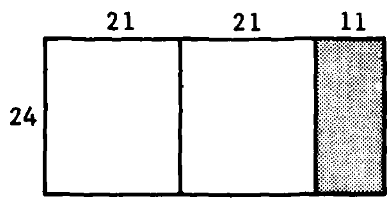
The box seed is 24 X 21. Apply the row completion procedure.

Step 1:

a = 24; b = 21.

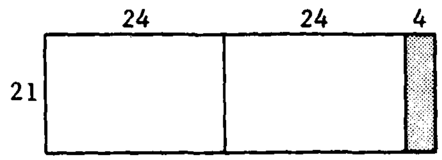
Step 2:

Try to generate a type 1 uniform complete row of height 24. First iteration of integer linear combination technique gives $n_1 = 2$, $n_2 = 0$ but the row has excessive waste.



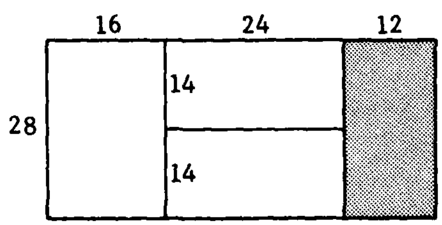
Row waste = $\frac{11}{53}$ X 100 = 21.2%.

Interchanging a and b gives $n_1 = 2$, $n_2 = 0$. The following acceptable complete row is generated and saved on list of complete rows.



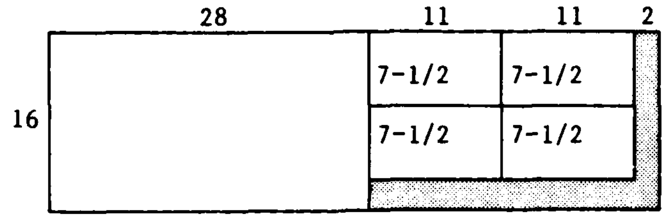
Row waste = $\frac{4}{52}$ X 100 = 7.69%.

The next available seed is 28 X 16. Attempts to obtain either a type 1 or type 2 uniform complete row by applying steps 1 and 2 of row completion procedure fail; likewise, an attempt to form a type 1, non-uniform row in step 3a fails. Step 3b gives a non-uniform row of height 28 by selecting the 14 X 24 size box and generating a column by a combination of 14-inch dimensions and the complete row by combination of the 16-inch dimension of the seed and 24-inch dimension of the selected box. The row is shown here, but its waste is excessive.



Row waste = $\frac{12}{52}$ X 100 = 23.1%.

Interchanging a and b, so that a = 16, b = 24, gives a successful type 2 non-uniform complete row which is put on list of complete rows.

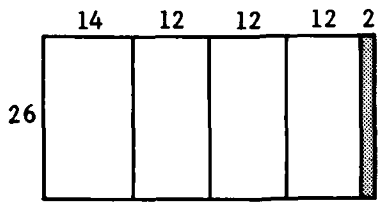


Column waste = $\frac{1}{16}$ X 100 = 6.25%.

Row waste = $\frac{2}{52}$ X 100 = 3.85%.

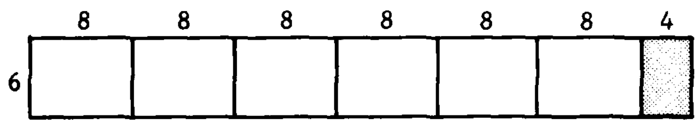
Since only one of the two available 28 X 16 boxes has been used, the next available seed is once again 28 X 16. Sufficient 11 X 7-1/2 boxes are available for immediate repetition of the last row cluster, and this row is placed on the list of complete rows.

Next available seed is 26 X 14. Formation of a uniform complete row fails, but step 3a of the row-completion procedure provides an acceptable type 1 non-uniform complete row of height 26 inches by combining the 14-inch dimension of the seed and the 12-inch dimension of the 26-inch X 12-inch size box selected by the integer linear combination technique.



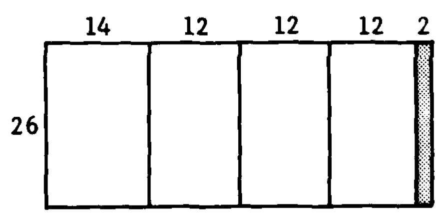
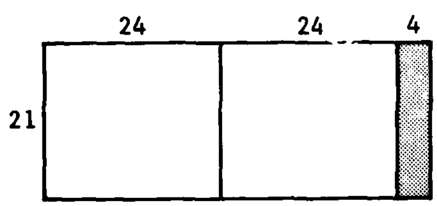
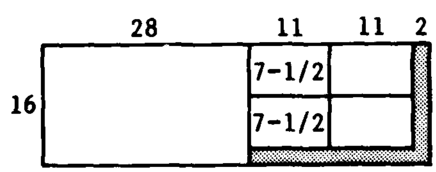
Row waste = $\frac{1}{52}$ X 100 = 3.85%.

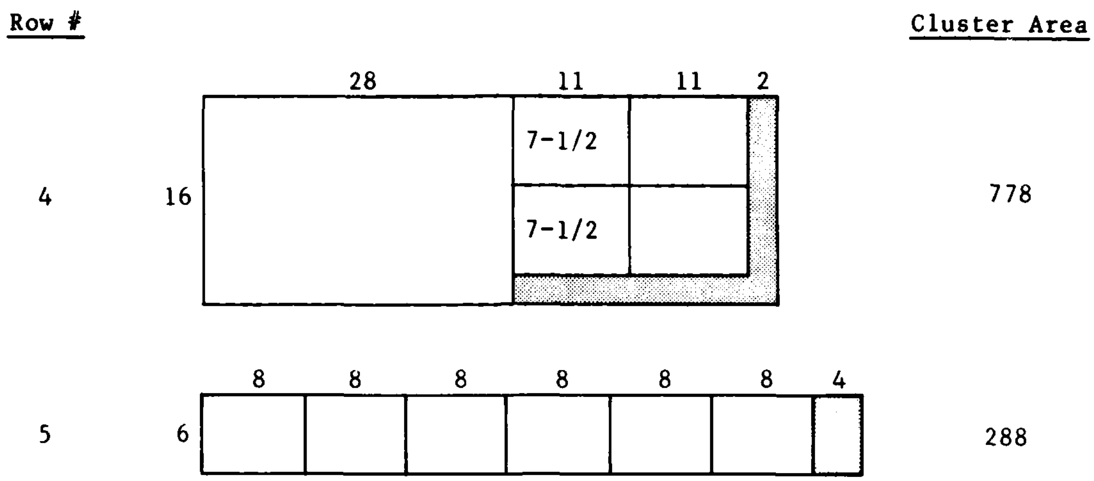
The box list now consists only of 8 X 6 boxes. Using a seed of 8 X 6, a uniform type 1 complete row is quickly obtained to exhaust the box list for group 2.



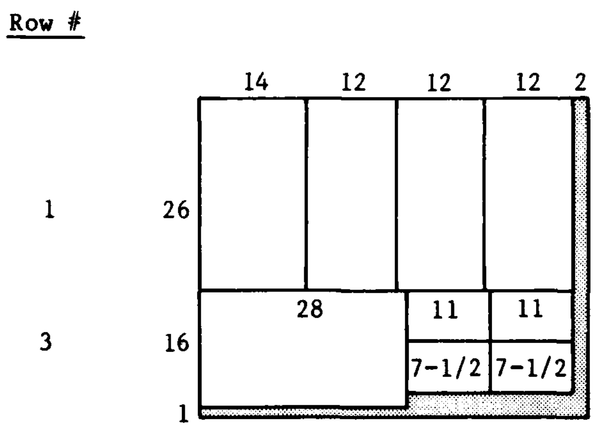
Row waste = $\frac{4}{52}$ X 100 = 7.69%.

The row list now consists of five complete rows, rearranged by decreasing row height as follows:

Row #		Cluster Area
1		364 + 3(312) = 1300
2		1008
3		778



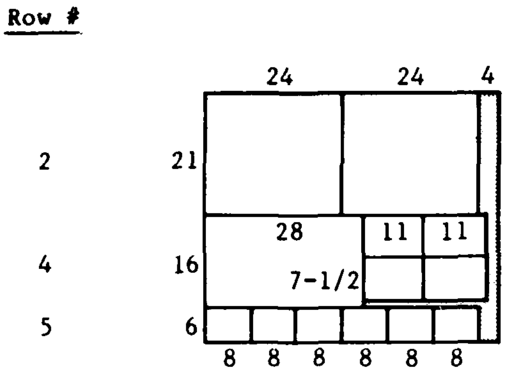
The row cluster seed has height 26 inches. Using the type 1 configuration formation procedure gives the following combination of rows 1 and 3:



PATTERN NO. 2

Pattern efficiency = $\frac{1300 + 778}{43 \times 52} \times 100 = 92.9\%$.

The next row cluster seed has row height = 21. The type 1 procedure can not produce a pattern with acceptable waste, but step 1 of the type 2 procedure immediately provides a satisfactory pattern.



PATTERN NO. 3

$$\text{Pattern efficiency} = \frac{1008 + 778 + 288}{43 \times 52} \times 100 = 93.2\%.$$

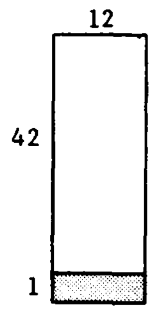
Group 2 is finished.

Group 3 Processing

Group 3 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
14	28 X 22	2
	42 X 12	1
	37 X 12	2
	18 X 14	2
	15 X 15	4

Seed is 28 inches X 22 inches. All efforts to obtain any type of complete row or complete column fail. Retain the 28 X 22 boxes and

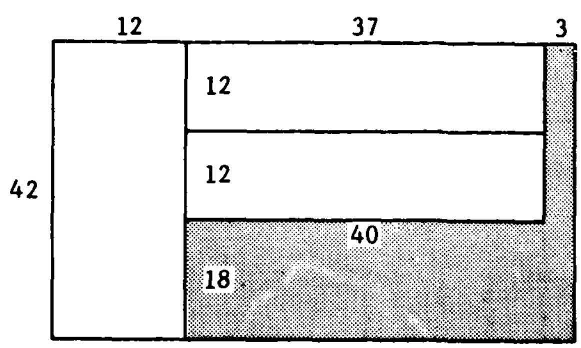
use the next available seed, 42 X 12. Again, no acceptable complete row can be obtained, but the following acceptable type 1 uniform complete column is obtained and is saved on the list of complete columns.



Column waste = $\frac{1}{43}$ X 100 = 2.3%.

Next seed is 37 inches X 12 inches. No satisfactory complete rows or columns can be constructed and the 37 X 12 box is retained on the box list. The same situation occurs with the two remaining seeds on the list.

The column cluster has greater value than any individual box, but efforts to complete a pattern with the column cluster seed by the type 1 and type 2 procedures fail because there are not enough column clusters. Applying steps 1 and 3 of the type 3 procedure gives excessive corner waste.

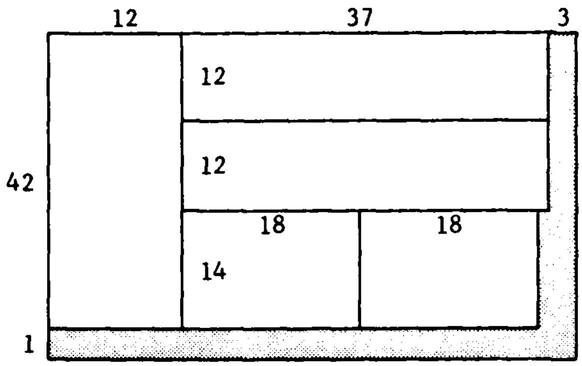


Corner waste =

$\frac{18 \text{ X } 40}{42 \text{ X } 40}$ X 100

= 42%.

Applying step 2 satisfactorily completes this pattern by the addition of partial columns.



Corner waste =

$$\frac{4 \text{ X } 14}{40 \text{ X } 14} \text{ X } 100$$

= 10%.

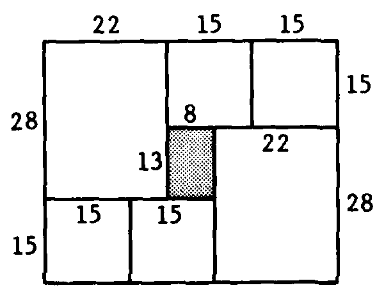
Pattern efficiency =

$$\frac{504 + 2(444) + 2(252)}{43 \text{ X } 52} \text{ X } 100$$

= 85%.

PATTERN NO. 4

Since the complete row and complete column lists are empty, the next seed is obtained from the box list and is 28 X 22. In the absence of complete rows and/or columns, the type 1, 2, and 3 *procedures all fail.* Applying the type 4 procedure gives a type 4A configuration which is successful (success measured by outerfill efficiency), requires no interior filling (fill measured by innerfill efficiency), and is a complete pattern (completeness measured by pattern efficiency).



PATTERN NO. 5

$$\text{Outerfill efficiency} = \frac{\text{actual load area including inner waste area}}{\text{maximum available load area}} \times 100$$

$$= \frac{(22 + 15 + 15) \times (28 + 15)}{43 \times 52} \times 100 = 100\%.$$

Outerfill waste = 100% - outerfill efficiency = 0.

$$\text{Hole waste} = \frac{\text{inner waste area}}{\text{actual load area including inner waste area}} \times 100$$

$$= \frac{8 \times 13}{(22 + 15 + 15)(28 + 15)} \times 100 = 4.65\%.$$

$$\text{Pattern efficiency} = \frac{\text{actual load area}}{\text{maximum pallet load area}}$$

$$= \frac{2(616) + 4(225)}{43 \times 53} \times 100 = 95.35\%.$$

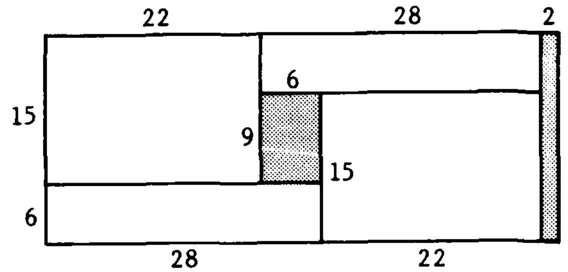
Group 3 is exhausted.

Group 4 Processing

Group 4 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
12	22 X 15	4
	28 X 6	4
	13-1/2 X 9	8
	11 X 11	8

No complete rows or columns can be formed using any of the boxes as a seed, and, consequently, type 1, 2, and 3 configurations cannot be

constructed. Applying the type 4 procedure with a seed of 22 X 15 gives a successful type 4B configuration.

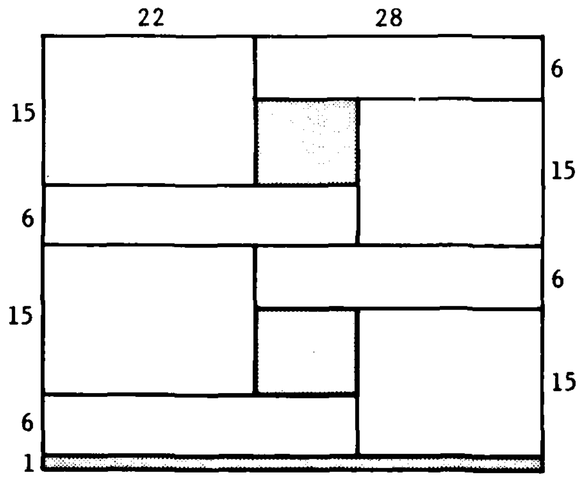


$$\text{Outerfill efficiency} = \frac{(28 + 22)(15 + 6)}{52(15 + 6)} \times 100 = 96.2.$$

Outerfill waste = 100 - 96.2 = 3.8%.

$$\text{Hole waste} = \frac{9 \times 6}{(28 + 22)(15 + 6)} \times 100 = 5.14\%.$$

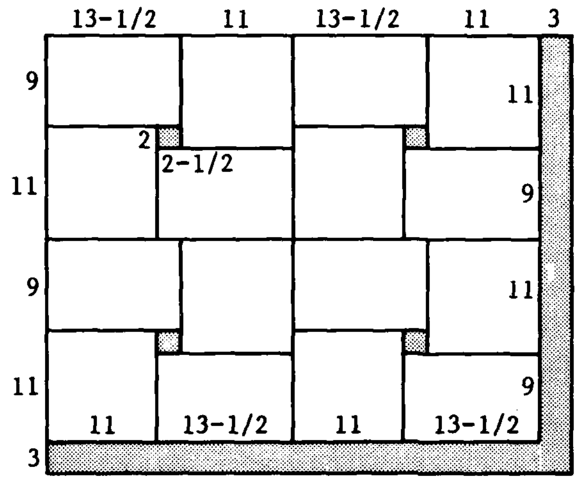
Repetition of the configuration leads immediately to a complete pattern.



PATTERN NO. 6

$$\text{Pattern efficiency} = \frac{2[2(15)(22) + 2(28)(6)]}{43 \times 52} \times 100 = 89.1\%.$$

Next seed is 13-1/2 X 9; type 4 procedure leads to a type 4C configuration which becomes, after repetition in two directions, a complete pattern.



PATTERN NO. 7

$$\text{Hole waste} = \frac{2(2\text{-}1/2)}{20(24\text{-}1/2)} \times 100 = 1.02\%.$$

$$\text{Outerfill efficiency} = \frac{2(13\text{-}1/2 + 11)\ 2(9 + 11)}{43 \times 52} \times 100 = 87.7\%.$$

Outerfill waste = 100 - 87.7 = 12.3%.

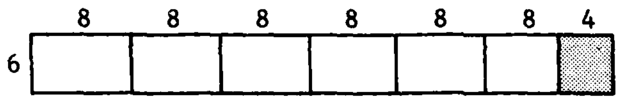
$$\text{Pattern efficiency} = \frac{8(121.5) + 8(121)}{43 \times 52} \times 100 = 86.8\%.$$

Group 4 is exhausted.

Group 5 Processing

Group 5 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
10	45 X 38	1
	22 X 11	4
	15 X 15	4
	8 X 6	6

Only one row may be completed and this occurs when the seed is 8 X 6; no complete columns are obtainable.

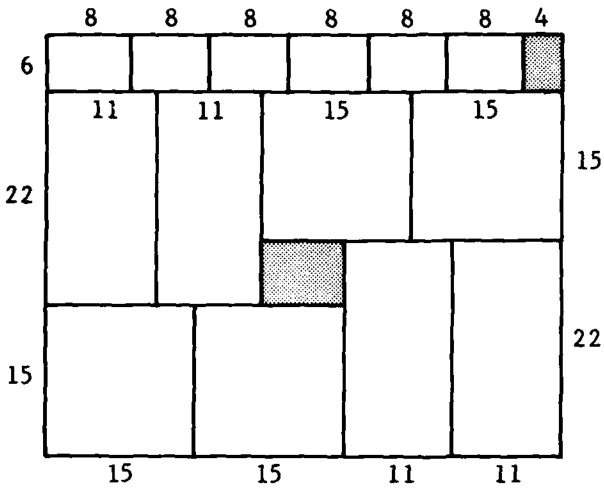


Row waste = $\frac{4}{52}$ X 100 = 7.69%.

Pattern completion with the type 1 and type 2 procedures fails because of insufficient complete rows or columns. Applying the type 3 procedure provides no acceptable partial column, so a complete pattern cannot be made. The incomplete pattern consists of only a single row cluster.

The next available seed that fits in the reduced available space, 37 inches by 52 inches, is a box 22 inches by 11 inches. Application of the type 4 procedure immediately leads to a successful type 4A

configuration which, combined with the type 3 row contribution, results in a complete pattern.



PATTERN NO. 8

$$\text{Pattern efficiency} = \frac{6(48) + 4(242) * 4(225)}{43 \text{ X } 52} \text{ X } 100 = 96.4\%.$$

The only item remaining in group 5 is the one 45- by 38-inch box. No acceptable cluster can be formed, so the 45- by 38-inch box is placed on the list of remaining boxes.

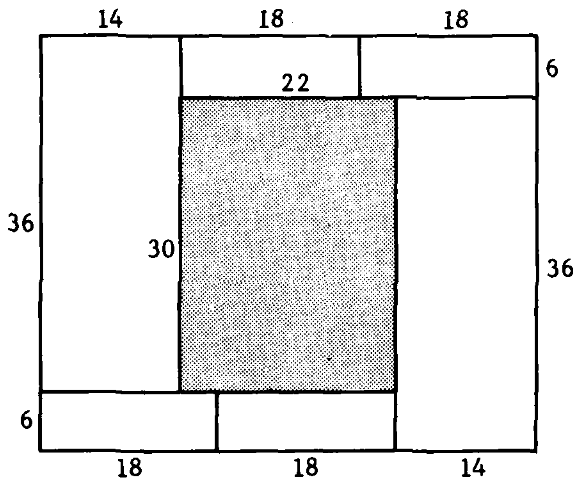
Group 6 Processing

This group is empty.

Group 7 Processing

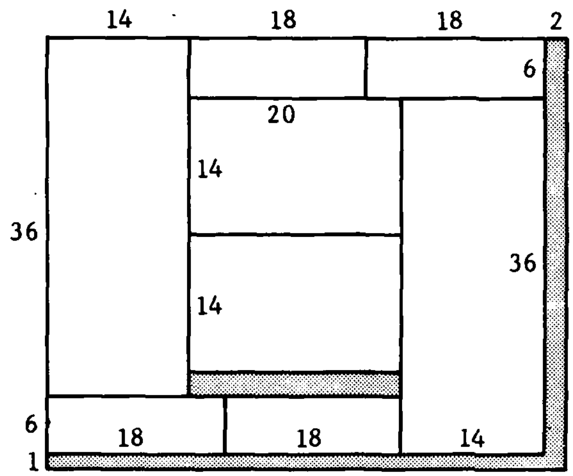
Group 7 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
8	36 X 14	2
	20 X 14	2
	18 X 6	4

No complete rows or columns are obtainable. Use of the 36- by 14-inch seed leads to a successful type 4A configuration with an incomplete pattern.



$$\text{Hole waste} = \frac{22 \times 30}{[14 + 2(18)][36 + 6]} \times 100 = 31.43\%.$$

An innerfill is required and is obtained by returning to the type 3 procedure and inserting a partial column composed of two 20- by 14-inch boxes.



PATTERN NO. 9

$$\text{Pattern efficiency} = \frac{2(504) + 4(1080 + 2(280)}{43 \text{ X } 52} \text{ X } 100 = 89.4\%.$$

Group 8 Processing

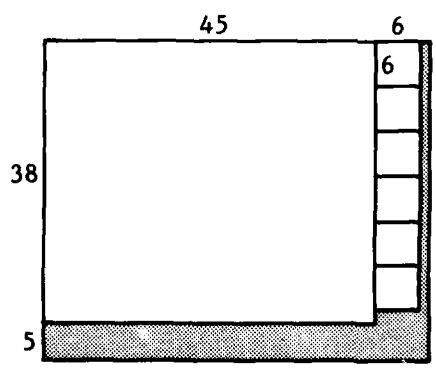
Group 8 Box List		
Group Height (Inches)	Box Lateral Dimensions L X W (Inches)	Quantity
6	6 X 6	6

There are too few boxes to form any acceptable cluster; these boxes are put on list of remaining boxes.

Remainde Group Processing

List of Remaining Boxes		
Remainder Group Height (Inches)	Box Dimensions L X W X H (Inches)	Quantity
10	45 X 38 X 10	1
	6 X 6 X 6	6

The seed box is 45 inches by 38 inches and a non-uniform type 2 row may be quickly constructed by the row completion procedure.



PATTERN NO. 10

Since no clusters or boxes remain, the row is used as an acceptable incomplete pattern with a flag denoting that the load population is exhausted.

Stacking and Completion

Ten patterns have been completed. For the relatively small number of boxes which have been arranged, no pattern repetition has occurred so that each unique pattern can be assigned to a tier. The height of a tier is equal to the height of the group from which the pattern was derived. Arranging the tiers in decreasing order of tier heights gives the following initial sequence: 18, 16, 16, 14, 14, 12, 12, 10, 10, 8. The following correlation for patterns, tiers, and tier heights is provided to monitor the composition of the stacks which result from tier rearrangement.

PATTERN NO.	TIER NO.	TIER HEIGHT
1	1	18
2	2	16
3	3	16
4	4	14
5	5	14
6	6	12
7	7	12
8	8	10
10	10	10
9	9	8

Container #1

Using the notation and formulas previously developed gives

$$HTCONT = 90''$$

$$HTFMAX = 48''$$

$$HTPALL = 6''$$

$$TOLPIL = 6'' \text{ (by assumption)}$$

$$TOLSTK = 3'' \text{ (by assumption),}$$

$$NMIN = \left| \frac{HTCONT}{HTFMAX} \right| = \left| \frac{90}{48} \right| = 2$$

The acceptable height range of full piles is given by:

$$HTFMAX - HTPALL - TOLPIL < HTPILE \leq HTFMAX - HTPALL$$

$$48 - 6 - 6 < HTPILE \leq 48 - 6$$

The full pile height range is (36, 42].

$$HTPMIN = HTCONT - (NMIN - 1) \times (HTFMAX - TOLPIL)$$

$$= 90 - (2 - 1) \times (48 - 6)$$

$$= 48$$

Since 48 > HTFMAX - TOLPIL, no parti piles can be formed.

When successive tier heights from the initial sequence are added until the addition of the next height would exceed the maximum permitted full pile height (42 inches), the following partitioning of the sequence into full piles and unused tiers is obtained:

18
16 } Unused Tiers since HTPILE = 18 + 16 = 34 $\leq$ 36

18 + 16 + 16 = 50 > 42

16
14 } Unused Tiers since 16 + 14 = 30 < 36

16 + 14 + 14 = 44 > 42

14
12
12 } Full Pile since 36 < 14 + 12 + 12 = 38 $\leq$ 42

10
10
8 } Unused Tiers since 10 * 10 * 3 = 28 $\leq$ 36

NMIN = 2
NFULL = 1
NPART = 0
NUSED = 9

Since NPART = 0, NFULL $\leq$ NMIN, Table 1 shows that the system is in state 4.

The state 4 stacking procedure forms a stack from all remaining full piles, in this instance only 1.

Initial Stack #1		List of Unused Tiers
14	Full Pile	18
12		16
12		16
		14
		10
		10
		8

AD-A120 343 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/4
CARGO LOADING--A PROPOSED APPROACH FOR MAXIMIZING SPACE UTILIZA--ETC(U)
SEP 82 G GERSTEL
UNCLASSIFIED DTNSRDC-82/099 NL

2 OF 2
AD A
120343

END
DATE
FILMED
11-82
DTIC

The height of the stack HSTACK is given by

$$\text{HSTACK} = \sum_{\substack{\text{all} \\ \text{I}}} \text{HTFULL(I)} + \text{NMIN X HTPALL}$$

$$= 14 + 12 + 12 + 2 \text{ X } 6 = 50$$

An initial underfill condition exists since HOVER = - HWASTE = HSTACK - HTCONT = 50 - 90 - -40. The wasted space in the stack is filled by adding unused tiers to the top of the stack until the addition of the next tier would overfill the stack. The stack then becomes

Modified Stack #1		List of Unused Tiers
16	Individual Tiers	16
18		14
14	Full Pile	10
12		10
12		8

HSTACK = 50 + 16 + 18 = 84

HWASTE = HTCONT - HSTACK = 90 - 84 = 6

Since stack tolerance TOLSTK = 3 and HWASTE > TOLSTK, additional stack modification is necessary.

The next stack refinement consists of trading the next entry on the list of unused tiers, if possible, for another tier in the stack to reduce the stack waste. The next entry is 16. Since the heights of all

individual tiers $\geq$ 16, the first pile beneath these tiers is considered. Since HWASTE = 6, the most desirable tier in the pile to trade for is a tier of height = 12. (If more than one tier has the same height, the uppermost tier will always be removed.) Before the trade is permitted, a check is made to ensure that the upper limit of the full pile height, 42, would not be exceeded. This check is satisfactory, and, therefore,

Modified Stack #1		List of Unused Tiers
16	Individual Tiers	14
18		10
14	Full Pile	10
16		8
12		12

HWASTE = 6 - (16 - 12) = 2 $\leq$ TOLSTK.

The stack is within tolerance and the final refinement in the completion stage may proceed. Insert a pallet under each full pile; begin at the topmost individual tier, add successive tiers without exceeding the maximum permitted full pile height 42, and insert a pallet under the newly formed pile.

Final Stack #1

16

18

pallet

14

16

12

pallet

$$HSTACK = HTCONT - HWASTE$$
$$= 90 - 2 = 88$$

The tiers on each pallet could be further analyzed to improve stability and weight but this is not required for this problem. The stack is complete and the next stack may be begun. The system is now in state 7, since NFULL = 0, NPART = 0, and NUNUSED = 5. The stacking strategy for this state is to immediately stack NMIN tiers, and to stack additional successive tiers from the list of unused tiers, one at a time, to fill the stack to its maximum permitted height without overfilling. The maximum permitted height (obtained by subtracting the height of NMIN pallets from the container height, HTCONT) is 90 - 2(6) = 78.

Initial Stack #1		List of Unused Tiers
10	Individual Tiers	10
14		8
		12

Modified Stack #2	List of Unused Tiers
12, 8, 10, 10, 14 — Individual Tiers	Empty

$$\text{HSTACK} = \sum_{\substack{\text{all}\\ \text{I}}} \text{HTTIER(I)} + \text{NMIN} \times \text{HTPALL}$$

$$= 12 + 8 + 10 + 10 + 14 + 2\ (6)$$

HSTACK = 64

HWASTE = HTCONT - HSTACK = 90 - 64 > TOLSK

No further improvement is possible since the load population is exhausted. Pallet insertion gives:

Final Stack #2

12

8

10

10

pallet

14

pallet

HSTACK = HTCONT - HWASTE

= 90 - 26 = 64

Container #2

$$HTCONT = 84''$$

$$HTFMAX = 36''$$

$$HTPALL = 6''$$

$$TOLPIL = 6'' \text{ (by assumption)}$$

$$TOLSTK = 3'' \text{ (by assumption)},$$

$$NMIN = \left| \frac{HTCONT}{HTFMAX} \right| = \frac{84}{36} = 3$$

For full piles,

$$HTFMAX - HTPALL - TOLPIL < HTPILE \leq HTFMAX - HTPALL$$

$$36 - 6 - 6 < HTPILE \leq 36 - 6$$

The full pile range is (24, 30].

For partial piles,

$$HTPMIN = HTCONT - (NMIN - 1) \times (HTFMAX - TOLPIL)$$

$$= 84 - (3 - 1) \times (36 - 6)$$

$$= 24 < HTFMAX - TOLPIL = 36 - 6 = 30$$

$$HTPMIN - HTPALL \leq HTPILE \leq HTFMAX - HTPALL - TOLPIL$$

$$24 - 6 \leq HTPILE \leq 36 - 6 - 6$$

The partial pile range is [18,24].

Partitioning the initial sequence of tier heights into full piles, partial piles, and unused tiers gives:

18 Partial Pile

16 Unused Tier

16
Full Pile
14

14
Full Pile
12

12
Partial Pile
10

10
Partial Pile
8

List of Full Piles	List of Partial Piles	List of Unused Tiers
16	18	16
14	12	
14	10	
12	10	
	8	
NFULL = 2	NPART = 3	NUNUSE = 1

NMIN = 3; Table 1 shows that the system is in state 3.

Initial Stack #1

18	Partial Pile #1	HTPART(1) = 18
14	Full Pile #1	HTFULL(1) = 14 + 12 = 26
12		HTFULL(2) = 16 + 14 = 30
16	Full Pile #2	
14		

$$HSTACK = HTPART(1) + \sum_{I = 1}^{NMIN - 1} HTFULL(I) + NMIN \times HTPALL$$

$$= 18 + 26 + 30 + 3 \times 6$$

HSTACK = 92

HOVER = HSTACK - HTCONT = 92 - 84 = 8

At this point the stack is overfilled and must be unloaded. The first pile that can be unloaded to produce a stack within tolerance is full pile #1, and the tier having height 12 inches is removed and added to the bottom of the list of unused tiers.

Modified Stack #1		List of Unused Tiers
18	Partial Pile #1	16
14	Full Pile #1	12
16, 14	Full Pile #2	

HWASTE = 8 - 12 = 4 > TOLSTK = 3

The next entry on the list of unused tiers = 16. Trading it for the 14-inch tier in full pile #1 gives

Modified Stack #1		List of Unused Tiers
18	Partial Pile #1	12
16	Full Pile #1	14
16, 14	Full Pile #2	

HWASTE = 4 - (16 - 14) = 2 $\leq$ TOLSTK = 3

The stack is within tolerance and, with a pallet under each pile, the final stack consists of:

<u>Final Stack #1</u>

18
pallet

HSTACK = HTCONT - HWASTE

= 84 - 2 = 82

16
pallet
16

14
pallet

The remaining lists are

List of Full Piles	List of Partial Piles	List of Unused Tiers
Empty	12	12
	10	14
	10	
	8	
NFULL = 0	NPART = 2	NUNUSE = 2

Table 1 now shows the system in state 6.

<u>Initial Stack #2</u>

10
8
Partial Pile #1

12
10
Partial Pile #2

HTPART(1) = 10 + 8 = 18

HTPART(2) = 12 + 10 = 22

$$HSTACK = \sum_{I=1}^{NMIN} HTPART(I) + NMIN \times HTPALL$$

$$= 18 + 22 + 3(6) = 58$$

$$HWASTE = HTCONT - HSTACK = 84 - 58 = 26$$

Filling the stack with unused tiers gives

$$\left.\begin{array}{r} 14 \\ 12 \end{array}\right\} \text{Individual Tiers}$$

$$\left.\begin{array}{r} 10 \\ 8 \end{array}\right\} \text{Partial Pile \#1}$$

$$\left.\begin{array}{r} 12 \\ 10 \end{array}\right\} \text{Partial Pile \#2}$$

$$HWASTE = 26 - 26 = 0$$

Inserting pallets gives

14
12
pallet   HSTACK = 84

10
8
pallet

12
10
pallet

Since there are three pallets in the stack and NMIN = 3, no pallet consolidation can be obtained.

SUMMARY OF STACKING RESULTS

Container #1	
<u>Stack #1</u>	<u>Stack #2</u>
Requires 2 pallets.	Requires 2 pallets.
Stack efficiency = $\frac{HSTACK}{HTCONT}$ X 100	Stack efficiency = $\frac{64}{90}$ X 100
= $\frac{88}{90}$ X 100 = 97.8%	= 71.1%
Container #2	
<u>Stack #1</u>	<u>Stack #2</u>
Requires 3 pallets.	Requires 3 pallets.
Stack efficiency = $\frac{82}{84}$ X 100	Stack efficiency = $\frac{84}{84}$ X 100
= 97.6%	= 100%

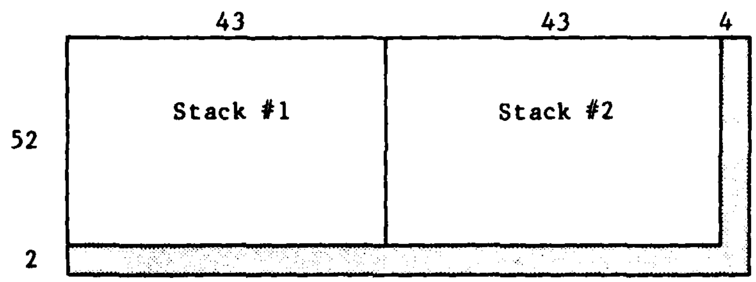
Container Loading

The available spaces in both containers have identical width and length dimensions.

Width = 90"

Length = 54"

Since the maximum pallet dimensions are 43" X 52", the following pattern is quickly obtained by first generating a type 1 uniform row which satisfies the type 1 configuration tolerance requirements.



DISCUSSION OF RESULTS

The cargo can be loaded in its entirety in either container #1 or container #2. Container #2 is, therefore, the proper choice, since the cost of shipment is lower. The more restrictive pallet loading height, 36", in pallet #2, requires the use of 6 pallets for loading the complete load instead of 4 pallets which would otherwise suffice. In certain instances, the extra space occupied by the additional pallets could result in a load which would not fit in the available space. For this problem, this extra space is available in container #2. The following comparison of space utilization efficiencies shows container #2 to be the more efficient.

Container #1

$$\text{Space Utilization Efficiency} = \frac{\text{Volume of Space Utilized}}{\text{Volume of Available Space}} \times 100$$

$$= \frac{\sum_{\text{all boxes}} \text{box volume} + \sum_{\text{all pallets}} \text{pallet volume}}{54 \times 90 \times 90}$$

$$= \frac{260580 + 46080}{453600} \times 100$$

$$= 70.1\%$$

Container #2

$$\text{Space Utilization Efficiency} = \frac{260580 + 69120}{54 \times 90 \times 84} \times 100$$

$$= 80.8\%$$

RECOMMENDATIONS AND CONCLUSIONS

The author recommends that a computer program be written to implement the proposed automated procedure for maximizing the space utilization of containers loaded with palletized loads. A candidate loading facility should be selected promptly to provide program inputs and to test and evaluate the proposed software.

Considerable testing will be required to determine how to trade off computer costs against improved space utilization efficiency. The sample problem was developed and solved to demonstrate how the various steps are interrelated and not to indicate any expected space utilization efficiencies. However, it is anticipated that, by adhering to a loading procedure which is systematic, logical, and without unwarranted iterations, and by using computers to do simple but laborious types of bookkeeping operations, a favorable balance between computer costs and space utilization efficiencies can be attained. The computer program could be a valuable analytic tool in making decisions, as the sample problem demonstrated. Additional benefits can also be expected from the comprehensive computer-generated reports to be produced at significant stages in the loading operation.

After initial program development and testing, the following enhancements merit consideration:

- Provide for varied loading orientation of boxes. The length or width of a box could be substituted for the height if the box were placed on its side.

- Extend program capability so that a promising pattern in a group can be completed, if necessary, by combining boxes in other groups to synthesize the height of the first group. In Figure 11, the pattern is completed by stacking boxes A and B or boxes C and D to obtain a "composite" box of the required group height.

- Provide for the construction of rows and columns in which the row height and column width exceed, within tolerance, the dimensions of the seed box. In Figure 12a, the result of the row construction technique in the initial program version is contrasted with that in Figure 12b where the row height can be enlarged within tolerance.

- Explore further the possibility of obtaining the program developed by Christofides and Whitlock[6] for dealing with loading situations in which only a limited number of boxes in a large variety of sizes is available. This program might also be beneficial for dealing with boxes for which successful patterns have not been found. Ideally, the proposed program could be interfaced with a program of the type developed by Christofides and Whitlock. The proposed program would be used when a large number of boxes is to be loaded, and the Christofides and Whitlock program would be used when a relatively small number of boxes of diverse and troublesome sizes remains.

Appendix C presents the flowcharts, Figures 13 through 32, for the proposed program loader.

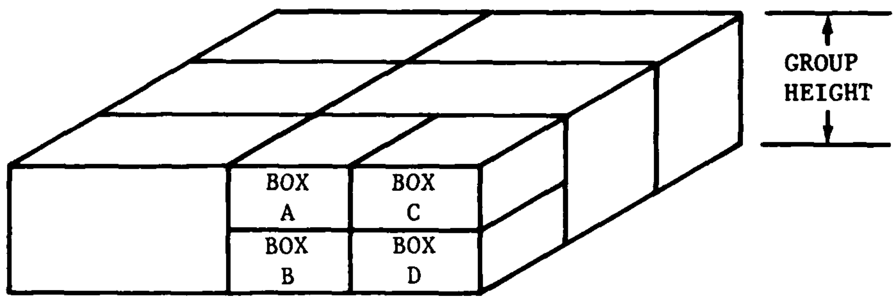
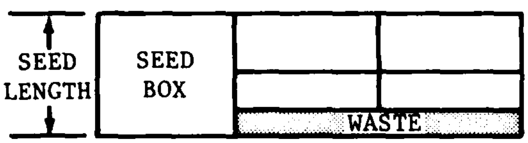
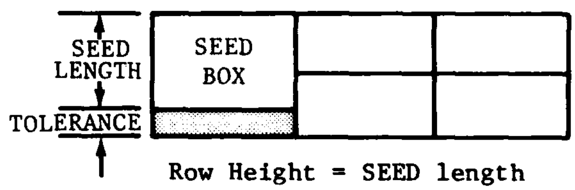


Figure - 11 Example of Pattern Completion by Synthesizing Group Heights



Row Height = SEED length

Figure 12a - Initial Program Development



Row Height = SEED length + tolerance

Figure 12b - Advanced Program Development

Figure 12 - Generalization of Row, Column Construction Techniques

ACKNOWLEDGMENTS

The author wishes to acknowledge Mr. Maurice Zubkoff for providing and explaining the background information related to some difficult, hard-to-find details of containerization, and also for his support and overall supervision of the project.

REFERENCES

1. Tabak, H.D., "Cargo Containers, Their Stowage, Handling and Movement," Cornell Maritime Press, Inc., Cambridge, Maryland, 1970.

2. NAVSEASYSCOM, "Containerization in Naval Logistics," Containerization Workshop, 25 - 27 June 1975, DTNSRDC, Carderock, Maryland.

3. Military Standard, "Palletized Unit Loads on 40" X 48" Pallets," MIL-STD-147B, 30 April 1968.

4. Gilmore, P.C. and R.E. Gomory, "Multistage Cutting Stock Problems of Two and More Dimensions," Op. Res., Vol. 13, No. 1, Jan. - Feb. 1965.

5. Herz, J.C., "Recursive Computational Procedure for Two-Dimensional Stock Cutting," IBM Journal of Research and Development, Vol. 16, No. 5, 1972.

6. Christofides, N. and C. Whitlock, "An Algorithm for Two-Dimensional Cutting Problems," Op. Res., Vol. 25, No. 1, Jan. - Feb. 1977.

7. Federal Supply Catalog, C8105/15-IL-N, FSC Class 8115, "Boxes, Cartons, and Crates," Defense General Supply Center, Virginia, 1 July 1968.

APPENDIX A
INDEX CHART FOR PALLET PATTERN DETERMINATION

INCHES IN LENGTH (columns) / INCHES IN WIDTH (rows)

	6	6½	7	7½	8	8½	9	9½	10	10½	11	11½	12	12½	13	13½	14	14½	15	15½	16	16½	17	17½	18	18½	19	19½	20	20½	21	21½
6	122	122	118	115	115	115	71	71	111	120		65	65	60	106	108	46	45	45	62	100	100	100	78	78	78	78	78	76	76	76	76
6½		121	121	67	67	67	24	61	61	120	65	65	38	63	63	75	75	62	62	58	58	58	58	47	47	19	19	119	42	42	42	42
7			117	62	59	114	24	110	110	116	49	49	49	105	75	75	75	75	39	99	99	99	99	19	19	19	19	42	42	42	42	42
7½					113	113	64	64	64	74	25	25	57	57	57	73	73	73			27	27	27	77	77	77	72	72	72	72	72	72
8					113	113	64	64	23	74	25	25	22	22	22	44	44		27	27	27	27	27	77	77	72	72	72	72	30	30	30
8½						113	74	74	109	112	25	25	22	22	22	44	44	27	27	27	27	27	98	72	72	30	30	30	30	30	30	30
9									108	108	48	21	21	21	20	18	18	15	15	15	15	14	97						26	26	26	
9½									108	108	48	21	20	20	20	18	18	14	14	14	14	14	97				26	26	26	26	26	26
10										108	21	21	20	20	17	17	17	14	14	14	14	14	97			26	26	26	26	26	26	36
10½											104	104	104	104	104	17	16	29	29	12	12	97	97	26	26	26	26	26	26	26		82
11																	28	28	28	11	11	9	9	26	26	26	26	8	8	8	102	102
11½															28	28	28	28	11	11	9	9	9	26	8	8	8	8	8	102	102	102
12															28	28	28	13	11	11	9	9	9	8	8	8	8	7	7	102	102	102
12½															28	28	13	11	11	9	9	9	9	8	7	7	7	7	102	102	102	102
13															13	103	103	11	11	9	9	9	9	7	7	7	7	7	102	102	102	102
13½																			37	37	37	96	96	7	7	7	7	5	5	5	5	5
14																		37	37	91	91	91	91	7	7	7	7	5	5	5	5	5
14½																															4	4
15																													4	4		
15½																													4	4	4	95
16																													4	95	95	95
16½																									4	4	4	95	95	95	95	95
17																									4	95	95	95	95	95	95	95
17½																																
18																																
18½																																
19																																
19½																																
20																																
20½																																
21																																6
21½																																
22																																
22½																																
23																																
23½																																
24																																
24½																																
25																																
25½																																
26																																
26½ THRU 34																																
34½																																
35																																
35½																																
36																																
36½																																
37																																
37½																																
38																																
38½																																
39																																
39½																																
40																																
40½																																
41																																
41½																																
42																																
42½																																
43																																

USE OF TABLES

1. **Determine length and width of container to nearest 1/2 inch.**
2. **Locate the length of container at the top, and width at left side of index chart, the container pattern number will be found at the intersection of two columns. *(Height may be substituted for width when* the containers are of sufficient strength to withstand superimposed loads, and where such placement will cause no adverse effect in the shipment or storage of the materiel.)**

APPENDIX A (Continued)

INCHES IN LENGTH

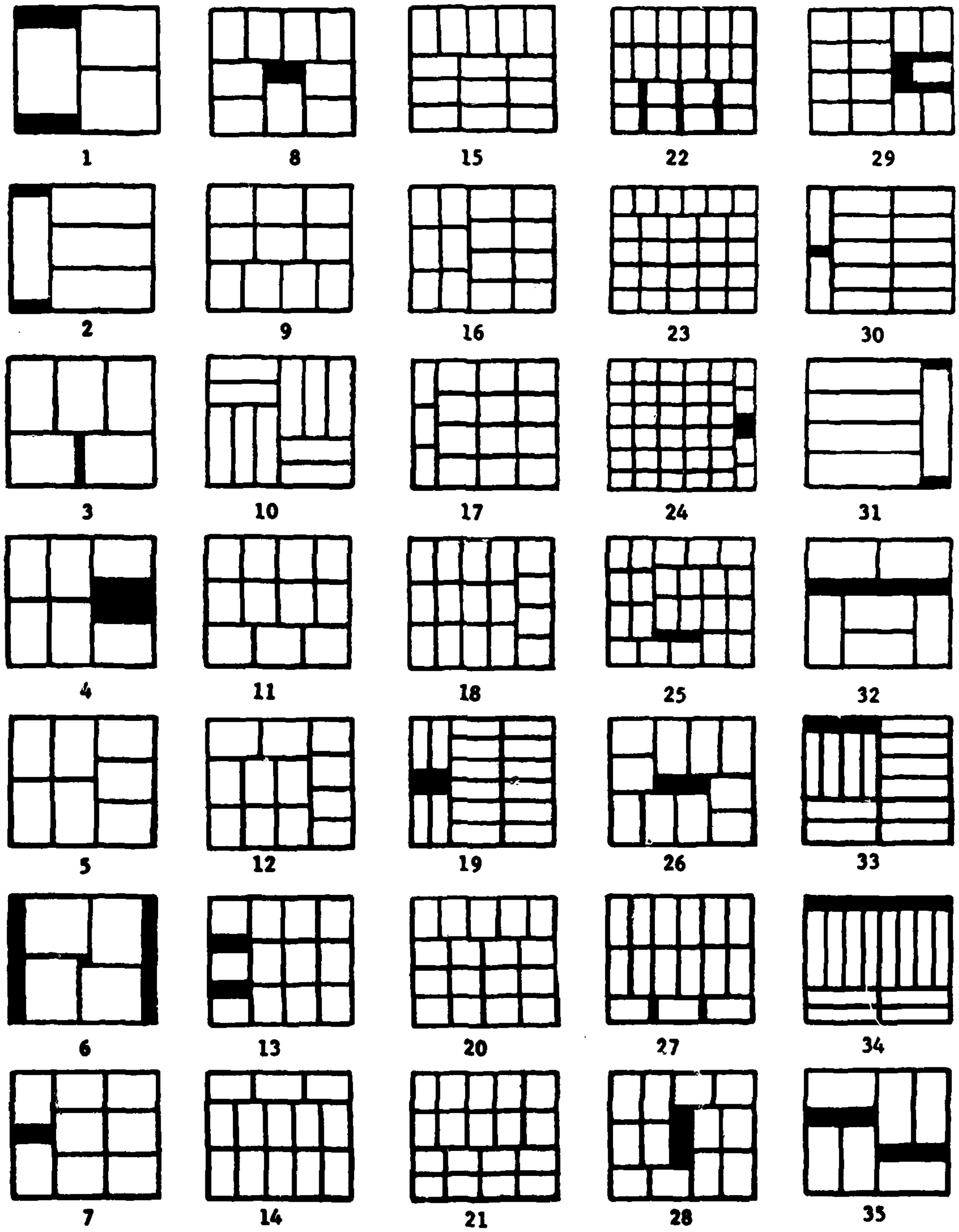
INCHES IN WIDTH	22	22½	23	23½	24	24½	25	25½	26	26½	27	27½	28	28½	29	29½	30	30½	31	31½	32	32½	33	33½	34	34½	35	35½	36	36½	37	37½
6	43	43	56	56	56	56	34	34	34	81	81	[illegible]				10	10	10	70	70	70	70	70	70	70	55	55	55	55	55	55	124
6½	56	56	56	56	34	34	34	34	34			10	10	10	10	10	10		80	80	80	80	55	55	55	55	55	55	55	55	69	69
7	41	41	33	33	33	93	93	93	93	10	10	10	10	10	10	80	80	80	80		54	54	54	54	54	54	54	54	54	69	69	69
7½	40				10	10	10	10	10	10	10	10	10												68	68	68	68	68	68	68	50
8	40	10	10	10	10	10	10	10	10	10	10														68	68	68	68	68	50	50	50
8½	10	10	10	10	10	10	10	10	10								68	68	68	68	68	68	68	68	68	68	50	50	50	50	50	50
9		36	36	36	36	36	36	92	92														35	35	53							
9½	36	36	36	36	92	92	92	92	92											35	35	35	35	53								
10	36	36	36	92	92	92	92	92	92								35	35	35	35	35	53	53						31	31	31	31
10½	92	92	92	92	92	92	92	92	92					35	35	35	35	35	35						31	31	31	31	31	31	31	31
11											35	35	35	35	35	35	35															
11½									35	35	35	35	35	35	35					79												
12								35	35	35	35	35	35				79	79	79													2
12½					35	35	35	35	35	35	35			79	79	79	79	79											2	2	2	2
13			35	35	35	35	35	35	35			79	79	79	79	79	79									2	2	2	2	2	2	2
13½	32	32	32	32	32	32	32	91	91														2	2	2	2	2	2	2	2	2	2
14	32	32	32	32	32	91	91	91	91												2	2	2	2	2	2	2	2	2	2	2	2
14½							3	3	3																							
15					3	3	3	3	3																							
15½			3	3	3	3	3	3	3																							
16		3	3	3	3	3	3	3	3																							94
16½	3	3	3	3	3	3	3	3	3																				94	94	94	94
17	3	3	3	3	3	3	3	3																			94	94	94	94	94	94
17½								90	90																1							
18							6	90	90															1	1							
18½						6	90	90	90													1	1	1								
19				6	6	90	90	90	90											1	1	1	1									
19½			6	90	90	90	90	90	90									1	1	1	1	1										
20		6	6	90	90	90	90	90	90								1	1	1	1	1											
20½	6	6	90	90	90	90	90	90	90						1	1	1	1	1	1												
21	6	90	90	90	90	90	90	90	90					1	1	1	1	1	1													
21½													1	1	1	1	1	1														
22																																
22½																																
23																																
23½																																
24																																89
24½																														89	89	89
25																													89	89	89	89
25½																											89	89	89	89	89	89
26																										89	89	89	89	89	89	89
26½ THRU 34																																
34½																																
35																																
35½																																
36																																
36½																																
37																																
37½																																
38																																
38½																																
39																																
39½																																
40																																
40½																																
41																																
41½																																
42																																
42½																																
43																																

APPENDIX A (Continued)

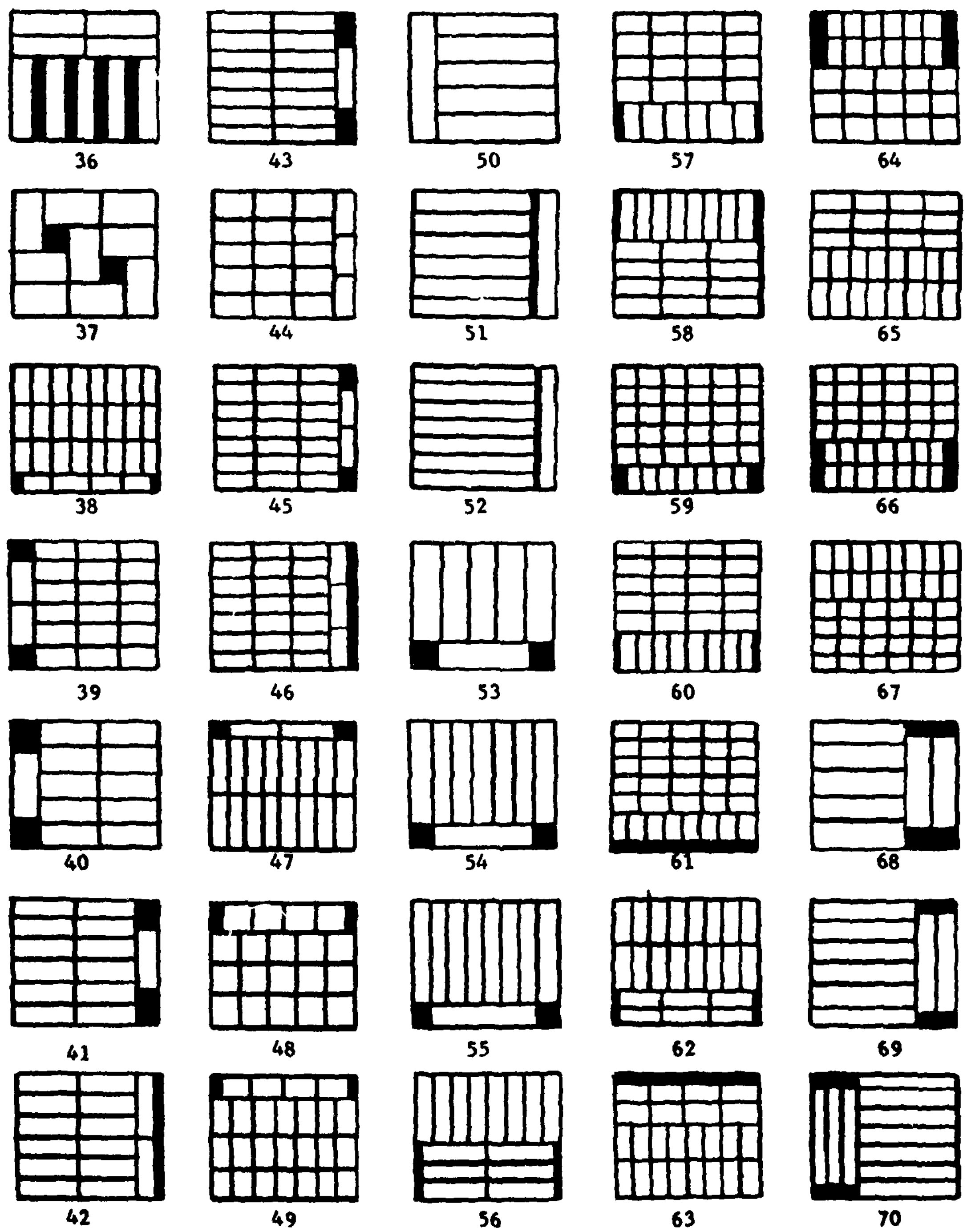
INCHES IN LENGTH (columns) / INCHES IN WIDTH (rows)

	38	38½	39	39½	40	40½	41	41½	42	42½	43	43½	44	44½	45	45½	46	46½	47	47½	48	48½	49	49½	50	50½	51	51½	52
6	124	124	124	124	124	52	52	52	52	52	52	123	123	123	123	123	123	123	123	123	123	123	123	123	123	123	123	123	123
6½	69	69	69	51	51	51	51	51	51	51	51						88	88	88	88	88	88	88	88	88	88	88	88	88
7	69	51	51	51	51	51	51	51	51	51	51	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88	88
7½	50	50	50	50	50	50	50	50	50	50	50										87	87	87	87	87	87	87	87	87
8	50	50	50	50	50	50	50	50	50	50	50				87	87	87	87	87	87	87	87	87	87	87	87	87	87	87
8½	50	50	50	50	50	50	50	50	50	50	50	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87
9					31	31	31	31	31	31	31														86	86	86	86	86
9½	31	31	31	31	31	31	31	31	31	31	107									86	86	86	86	86	86	86	86	86	86
10	31	31	31	31	31	31	31	31	31	107	107				86	86	86	86	86	86	86	86	86	86	86	86	86	86	86
10½	31	31	31	31	31	31	31	31			86	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86
11							2	101	101	101	101																		
11½			2	2	2	2	101	101	101	101	101																		85
12	2	2	2	2	2	101	101	101	101	101	101														85	85	85	85	85
12½	2	2	2	2	101	101	101	101	101	101	101										85	85	85	85	85	85	85	85	85
13	2	2	101	101	101	101	101	101	101	101	101						85	85	85	85	85	85	85	85	85	85	85	85	85
13½	2												85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
14	2										94	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
14½							94	94	94	94	94																		
15					94	94	94	94	94	94	94																		
15½		94	94	94	94	94	94	94	94	94	94																		
16	94	94	94	94	94	94	94	94	94	94	94																		
16½	94	94	94	94	94	94	94	94	94	94	94																		
17	94	94	94	94	94	94	94	94	94	94	94																		
17½																											84	84	84
18																									34	84	84	84	84
18½																						84	84	84	84	84	84	84	84
19																				84	84	84	84	84	84	84	84	84	84
19½																	84	84	84	84	84	84	84	84	84	84	84	84	84
20															84	84	84	84	84	84	84	84	84	84	84	84	84	84	84
20½													84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84
21											84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84
21½									89	89	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84	84
22							89	89	89	89	89																		
22½					89	89	89	89	89	89	89																		
23			89	89	89	89	89	89	89	89	89																		
23½	89	89	89	89	89	89	89	89	89	89	89																		
24	89	89	89	89	89	89	89	89	89	89	89																		
24½	89	89	89	89	89	89	89	89	89	89	89																		
25	89	89	89	89	89	89	89	89	89	89	89																		
25½	89	89	89	89	89	89	89	89	89	89	89																		
26	89	89	89	89	89	89	89	89	89	89	89																		
26½ THRU 34																													
34½																													83
35																											83	83	83
35½																										83	83	83	83
36																									83	83	83	83	83
36½																								83	83	83	83	83	83
37																						83	83	83	83	83	83	83	83
37½																					83	83	83	83	83	83	83	83	83
38																			83	83	83	83	83	83	83	83	83	83	83
38½																		83	83	83	83	83	83	83	83	83	83	83	83
39																	83	83	83	83	83	83	83	83	83	83	83	83	83
39½																83	83	83	83	83	83	83	83	83	83	83	83	83	83
40															83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
40½													83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
41												83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
41½											83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
42										83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
42½									83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83
43									83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83

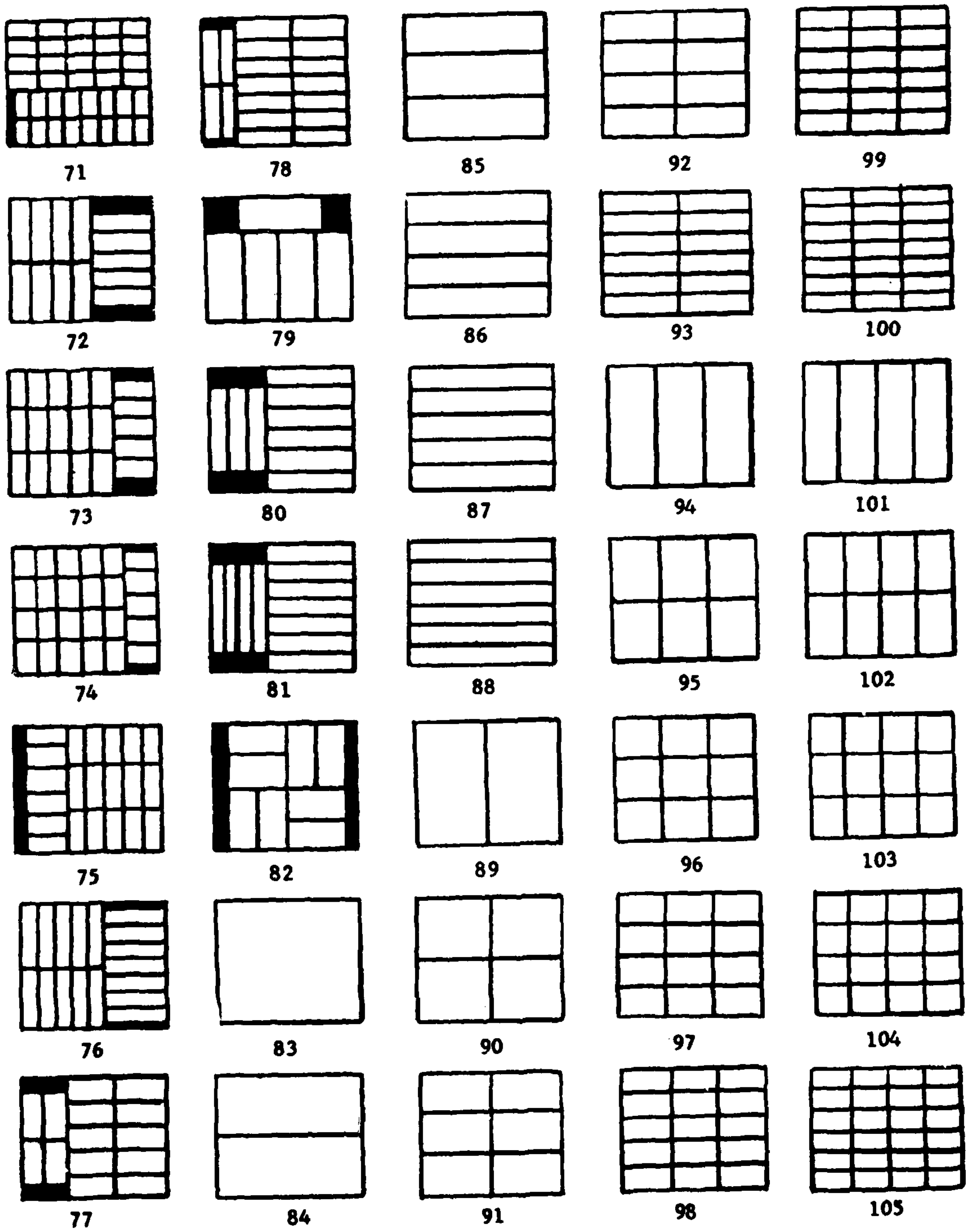
APPENDIX B
PALLET PATTERNS ON 40" x 48" PALLETS

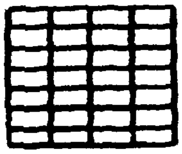


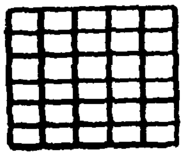
APPENDIX B (Continued)

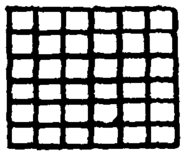


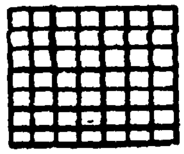
APPENDIX B (Continued)

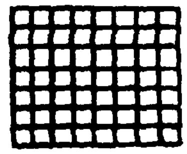


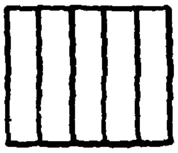

106

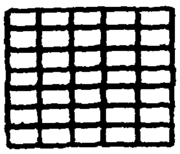

110

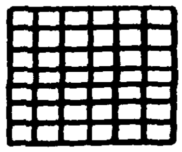

114

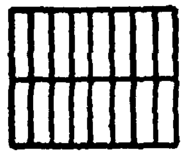

118

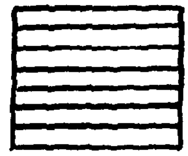

122

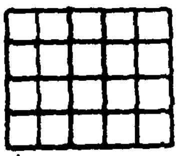

107

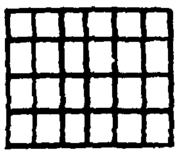

111

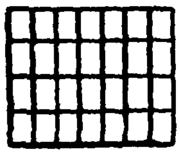

115

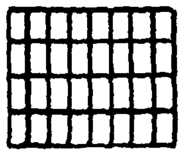

119

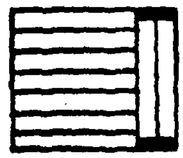

123

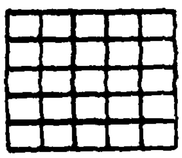

108

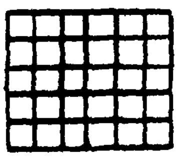

112

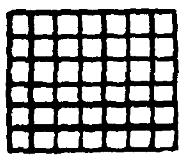

116

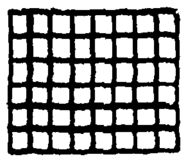

120


124


109


113


117


121

APPENDIX C - FLOWCHARTS OF PROPOSED PROGRAM LOADER
MAIN PROGRAM LOADER

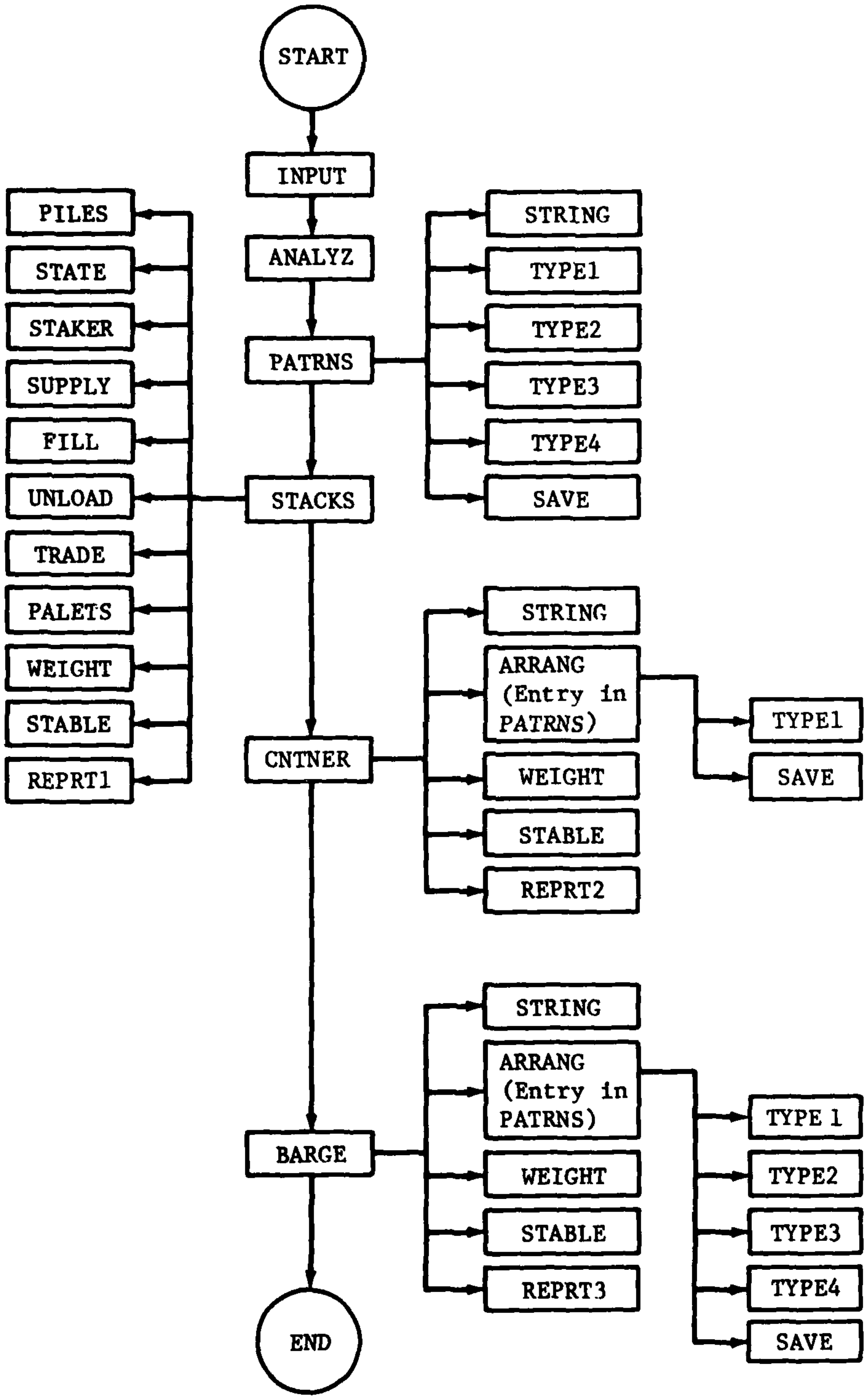


Figure 13 - Main Program Loader, Flowchart

SUBROUTINE INPUT

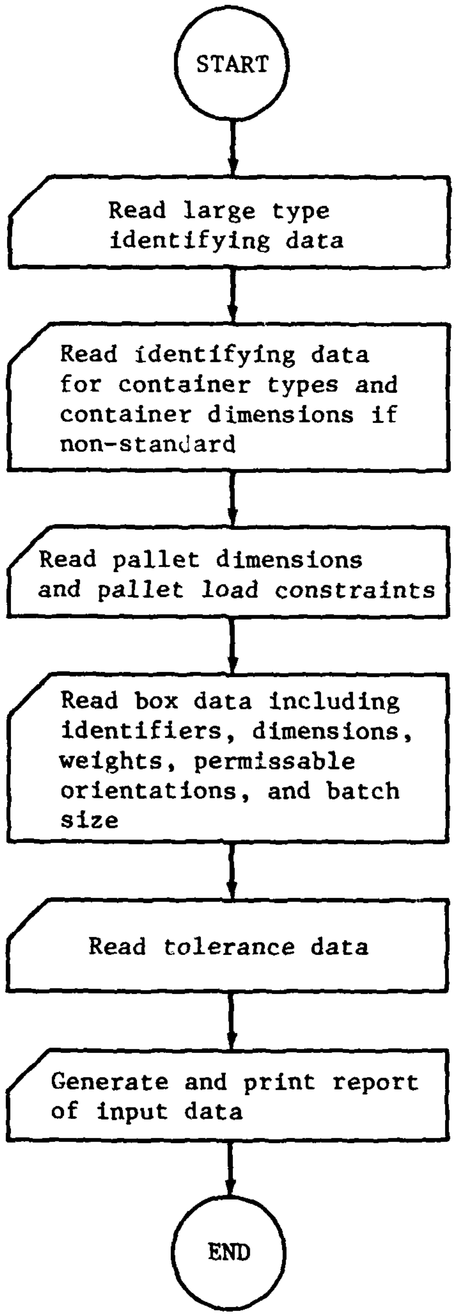


Figure 14 - Subroutine INPUT, Flowchart

SUBROUTINE ANALYZ

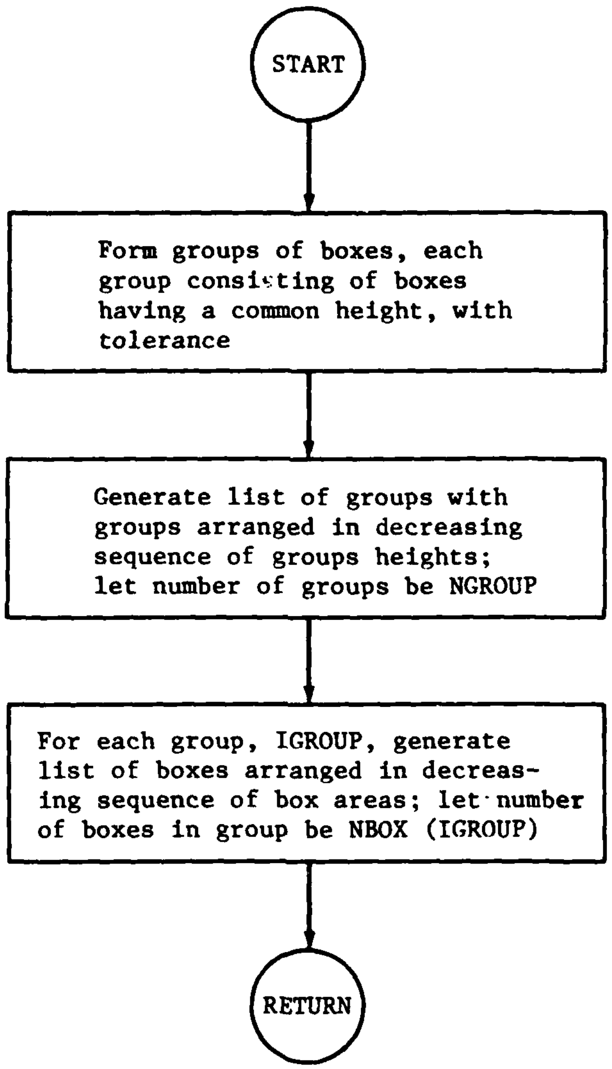


Figure 15 - Subroutine ANALYZ, Flowchart

Figure 16 - Subroutine PATRNS, Flowchart

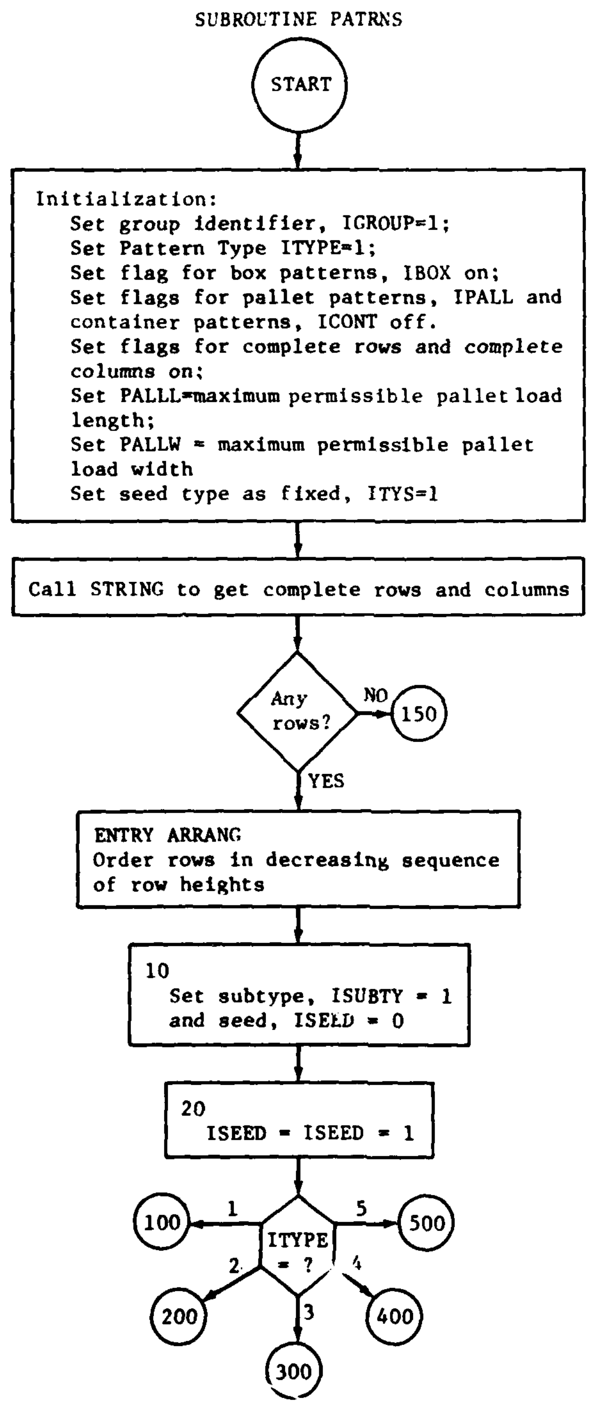


Figure 16 (Continued)

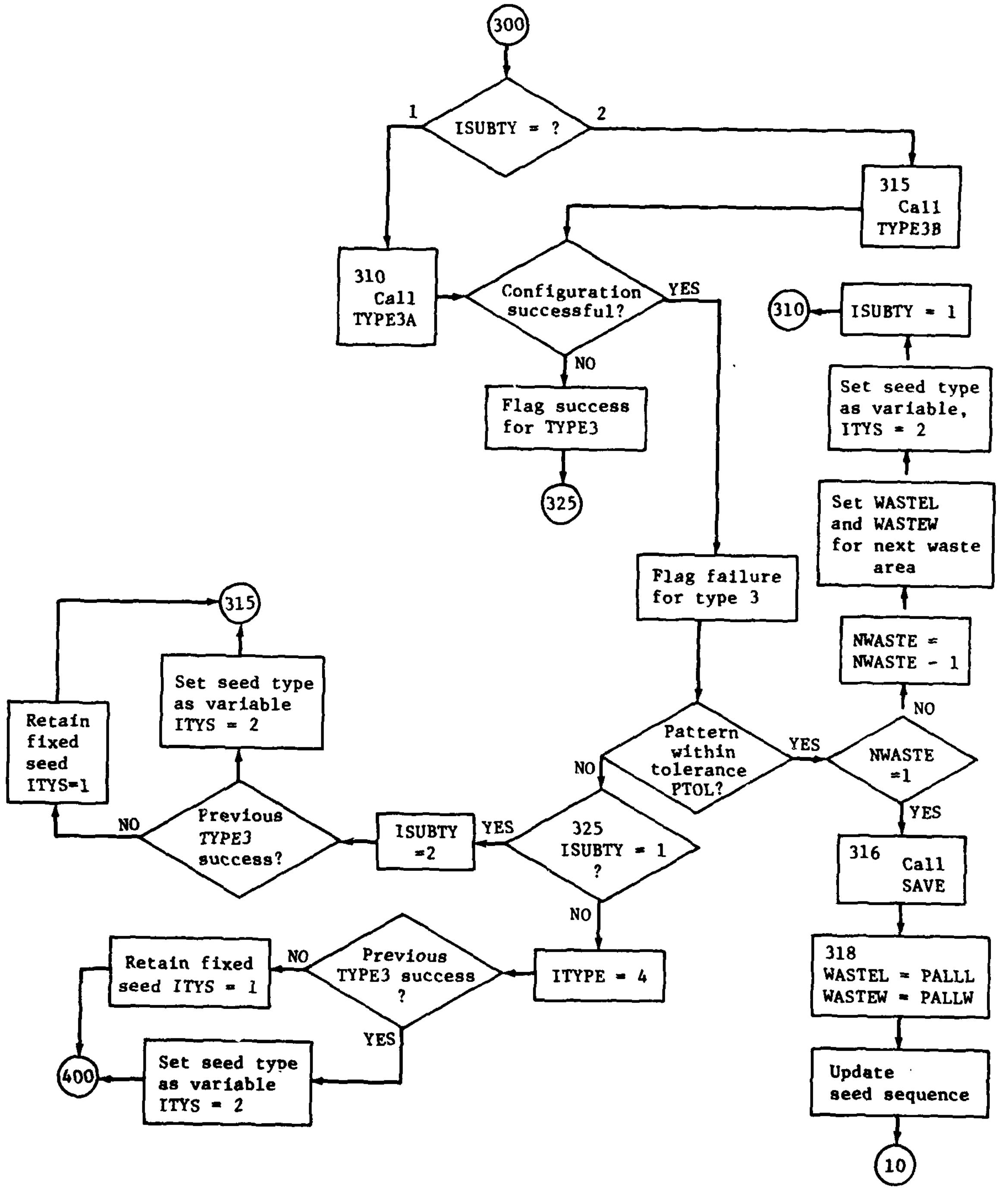


Figure 16 (Continued)

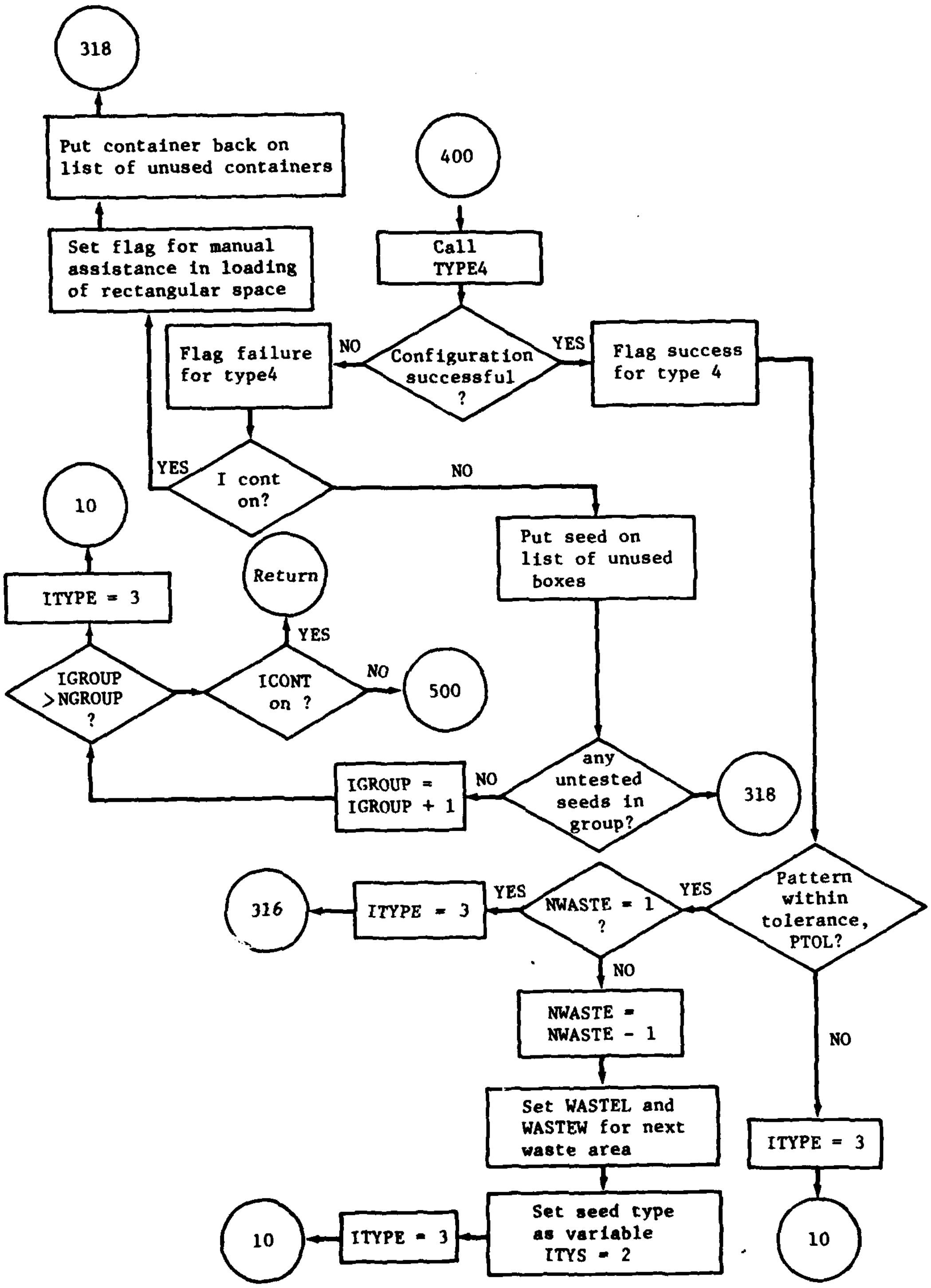


Figure 16 (Continued)

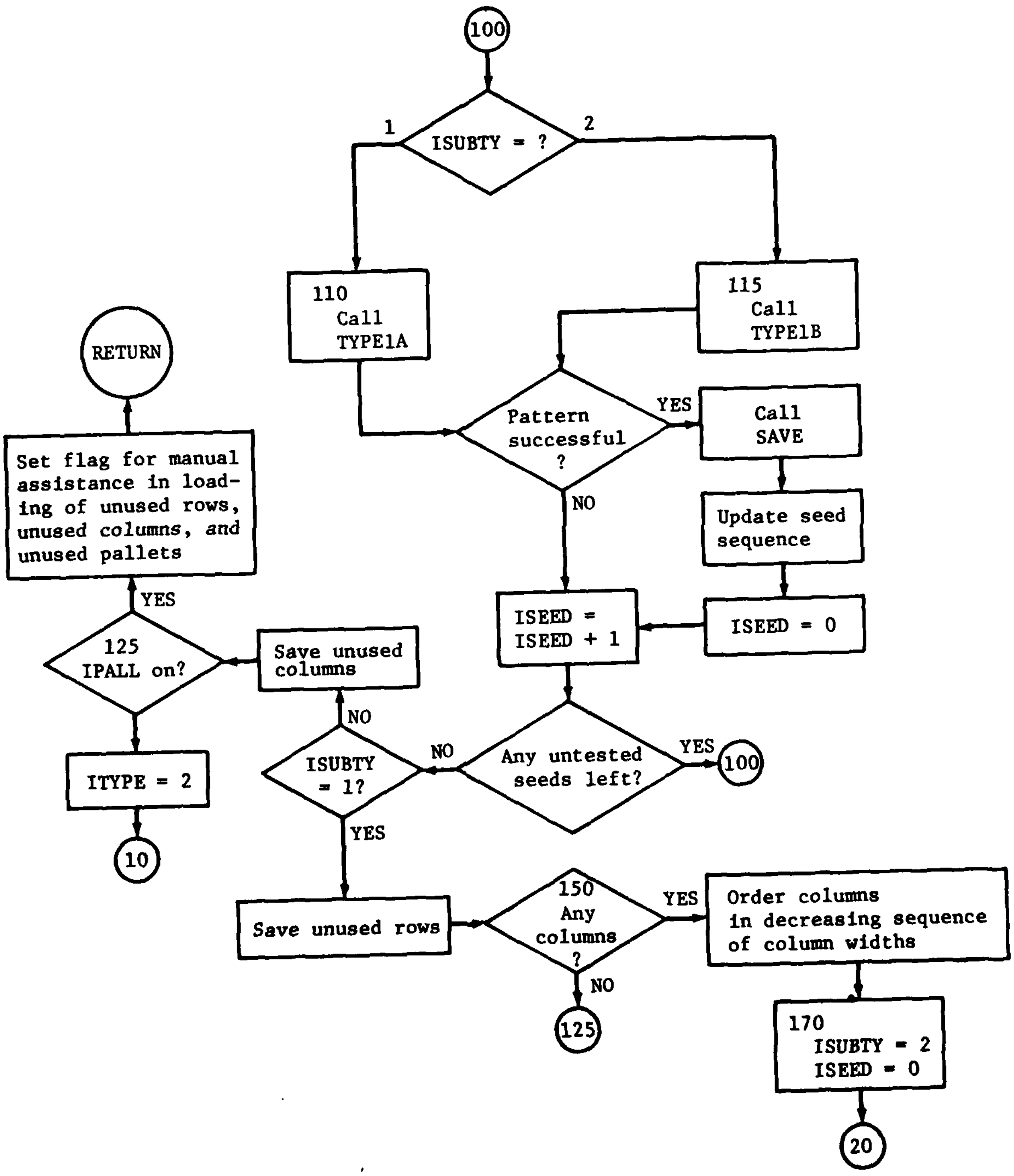


Figure 16 (Continued)

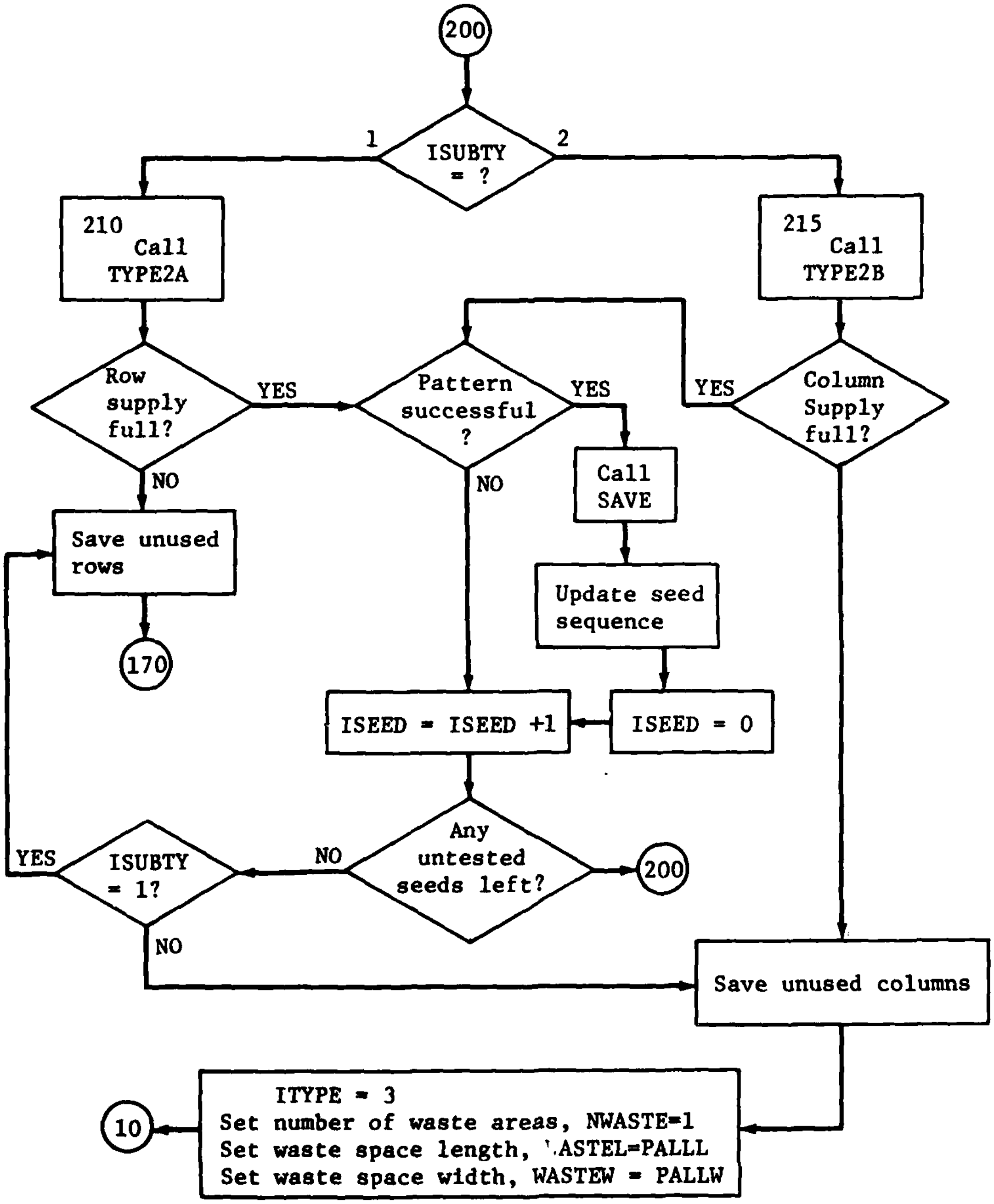


Figure 16 (Continued)

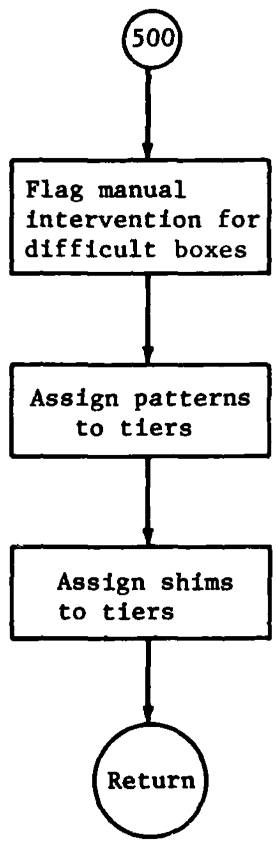


Figure 17 - Subroutine STRING, Flowchart

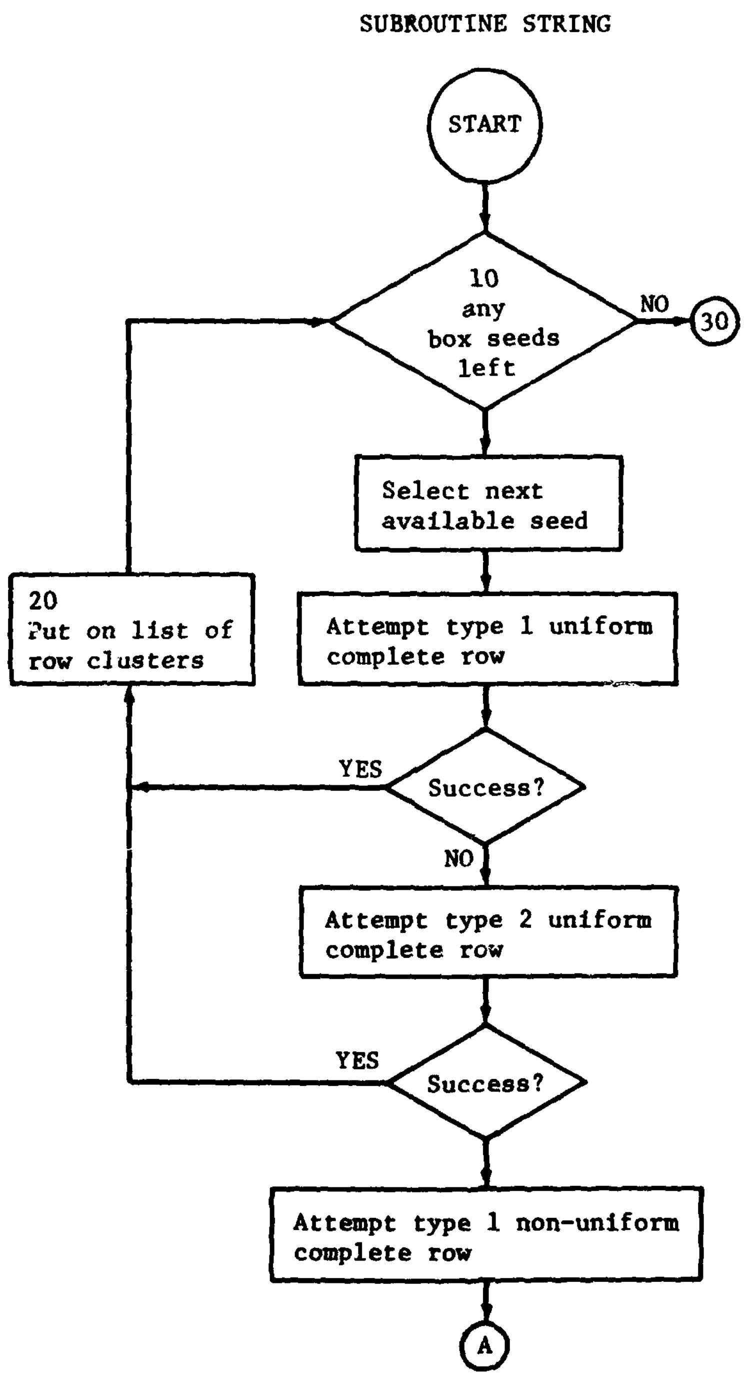


Figure 17 (Continued)

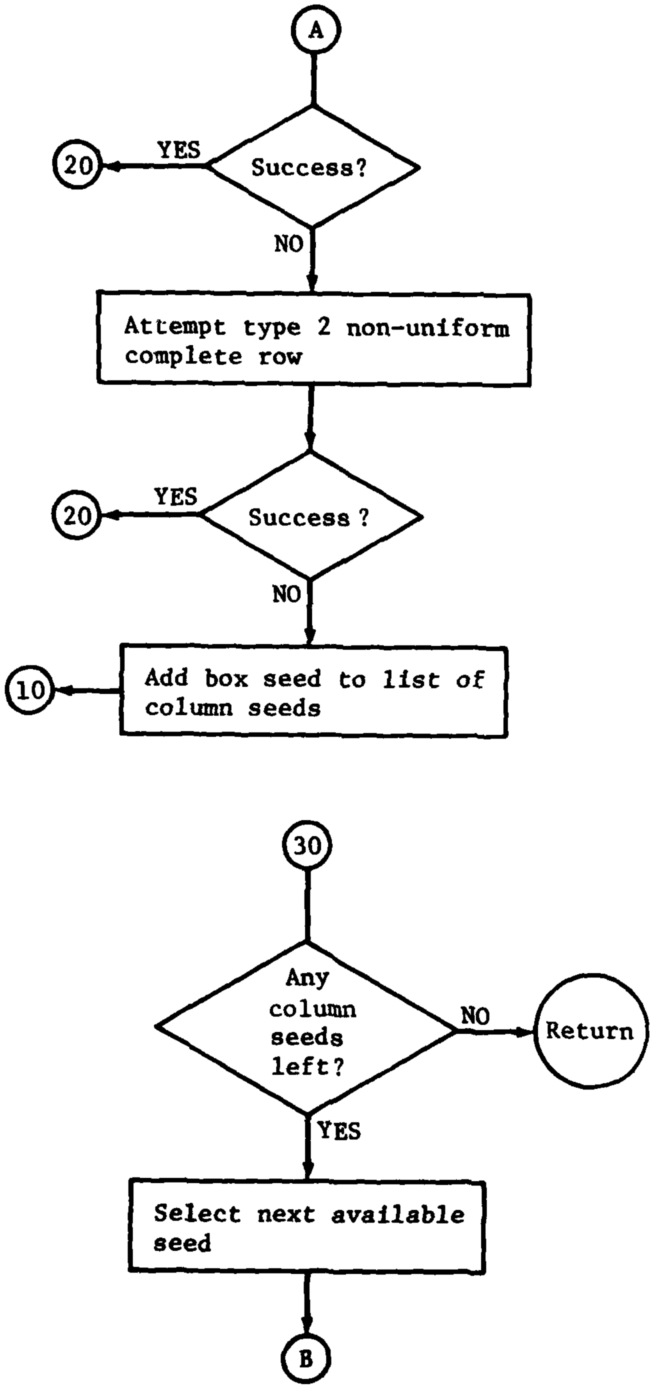
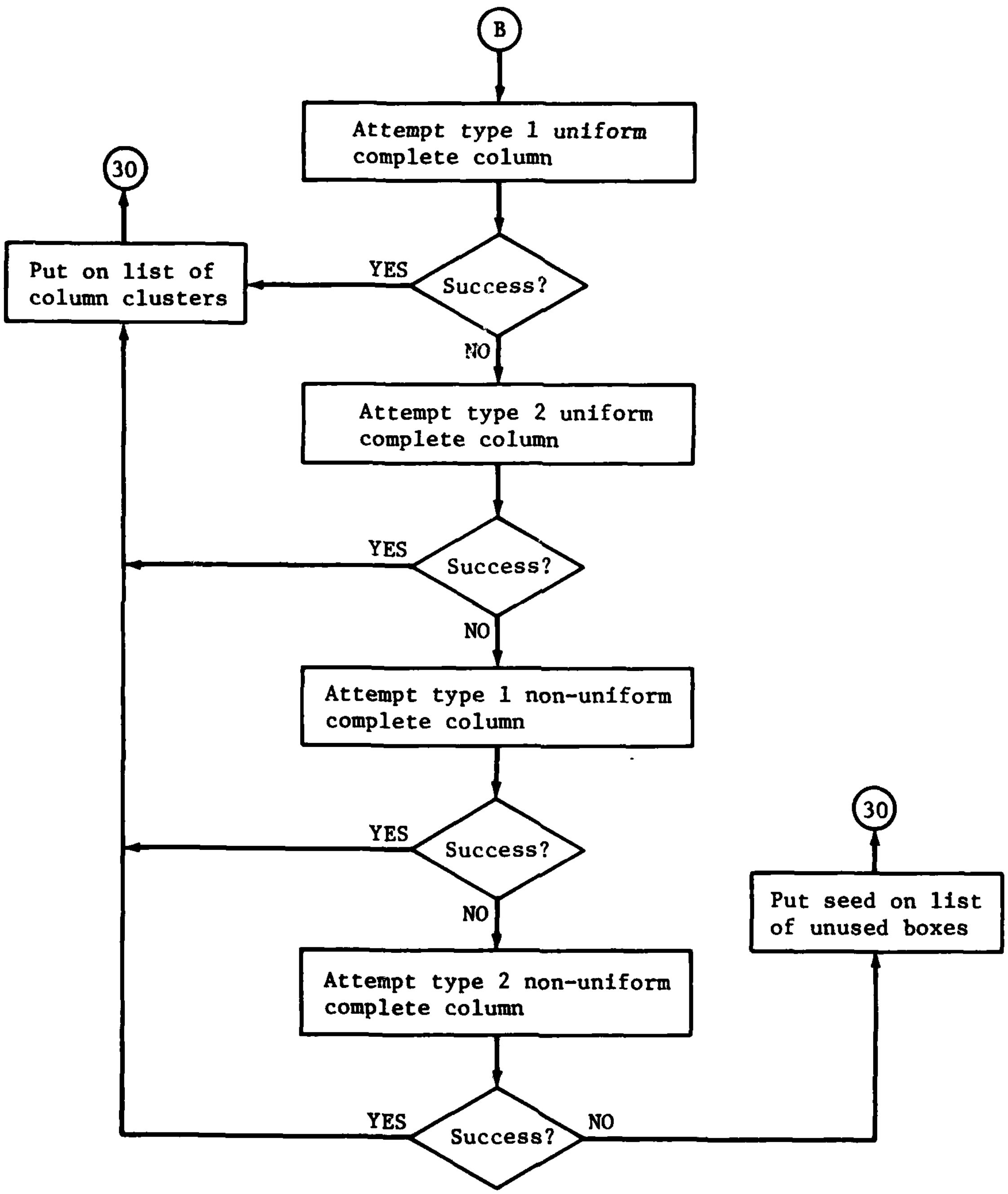
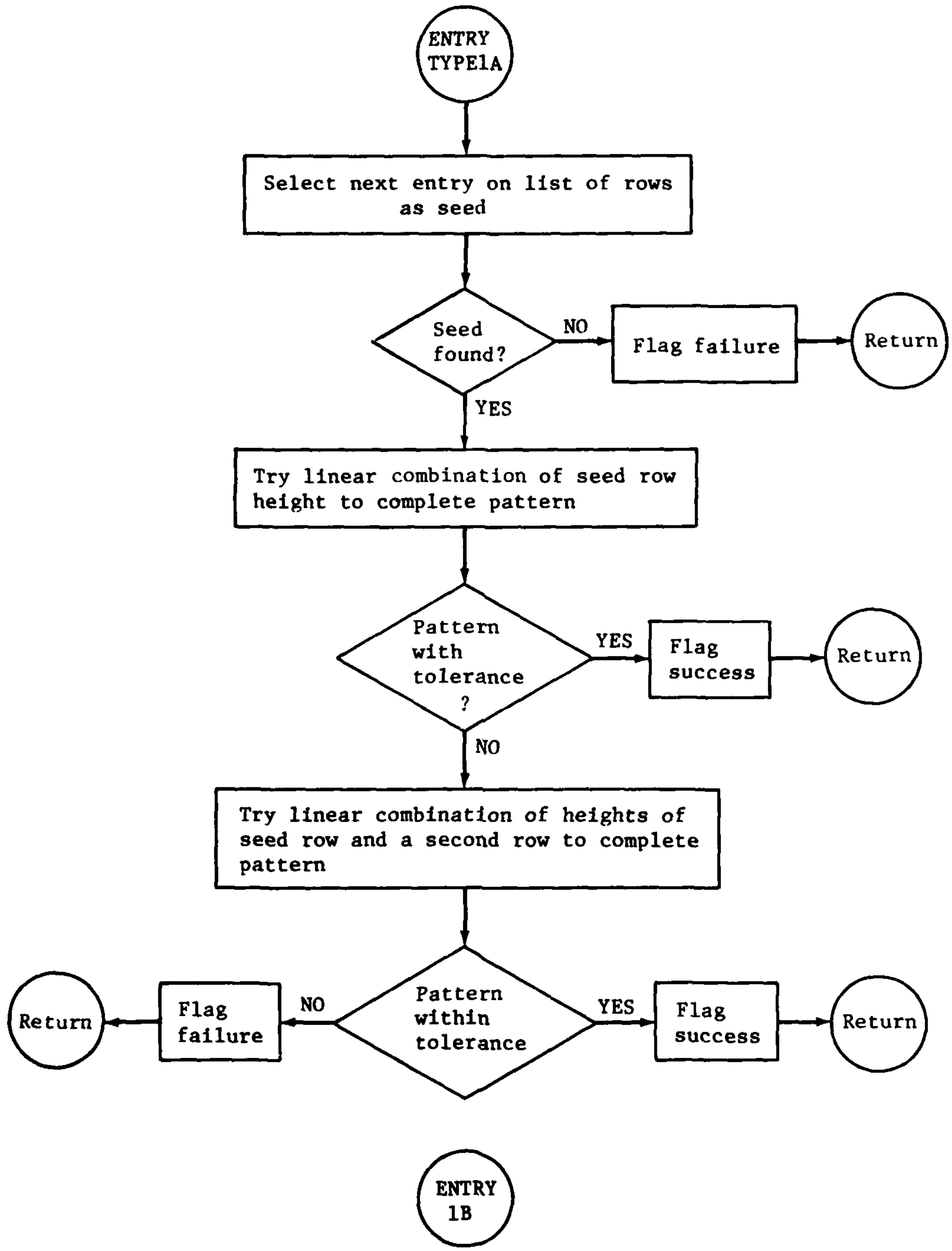


Figure 17 (Continued)



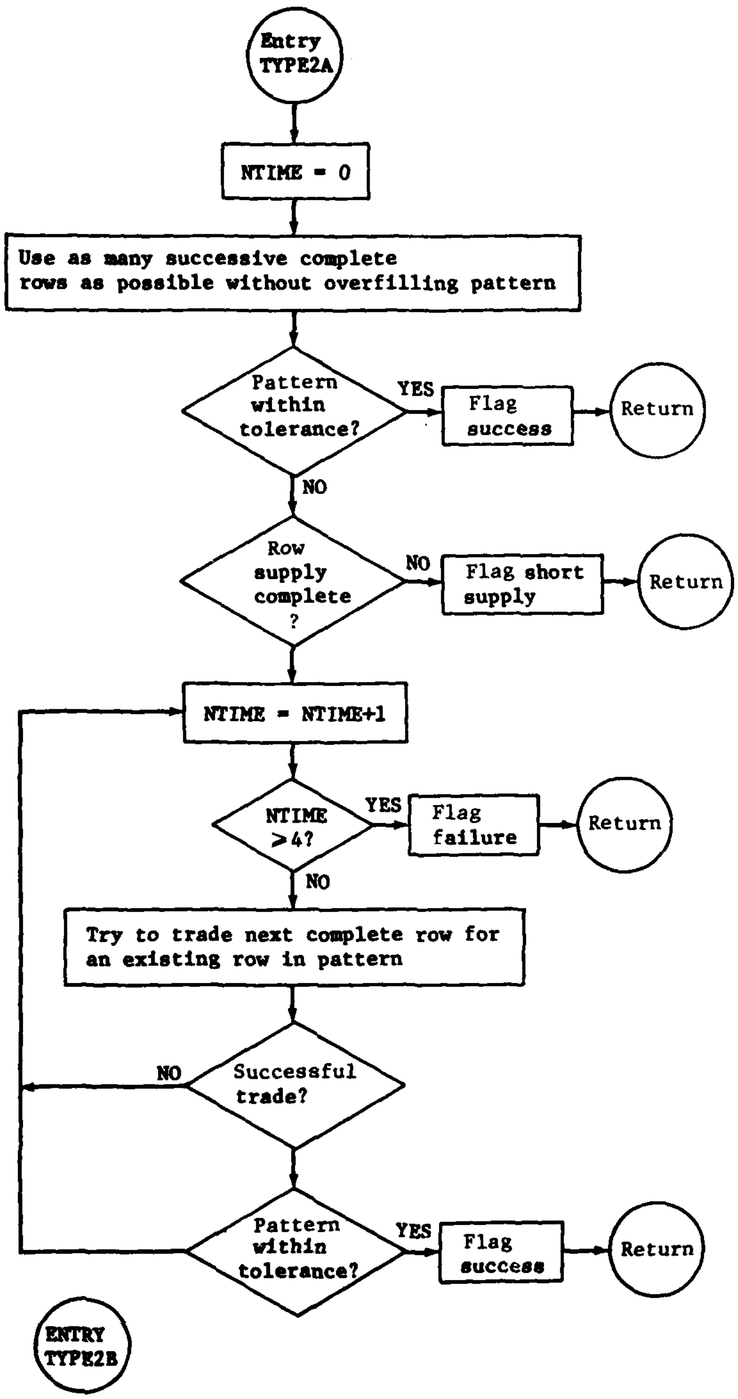
SUBROUTINE TYPE 1



Same flowchart as TYPE 1A except that column replaces row and column width replaces row height.

Figure 18 - Subroutine TYPE1, Flowchart

SUBROUTINE TYPE 2



Same flowchart as TYPE 2A except that columns are to be substituted for rows

Figure 19 - Subroutine TYPE2, Flowchart

Figure 20 - Subroutine TYPE3, Flowchart

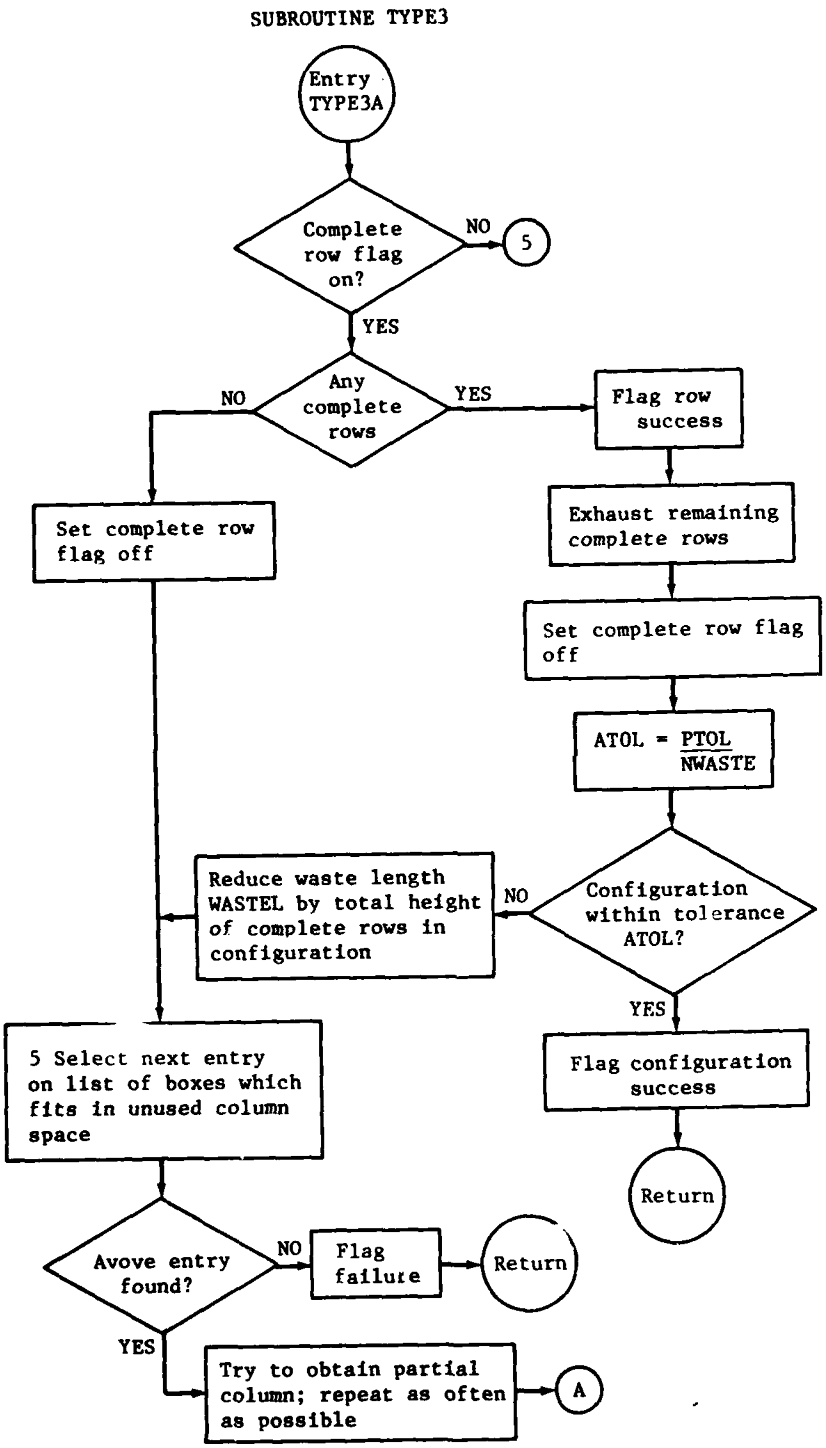


Figure 20 (Continued)

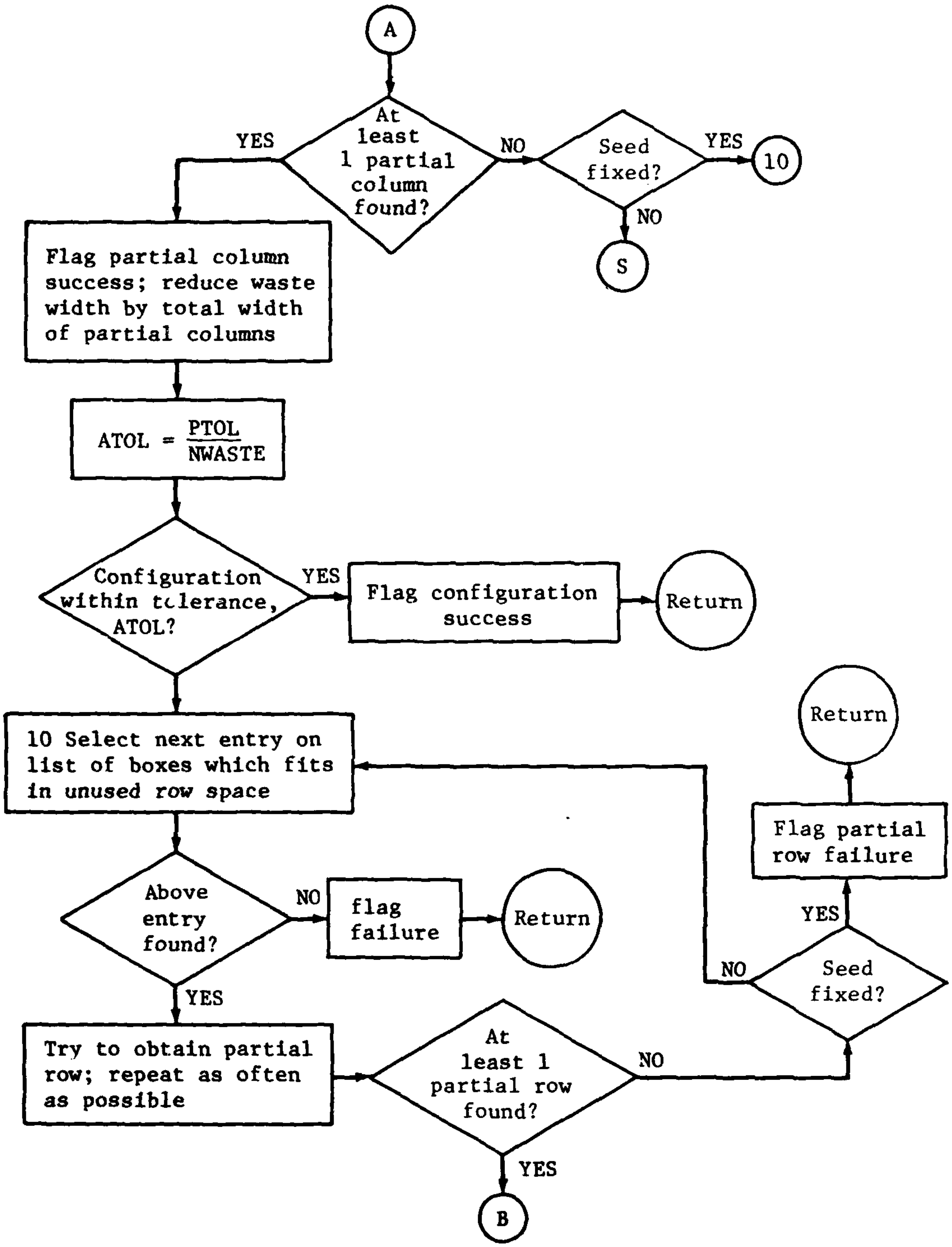
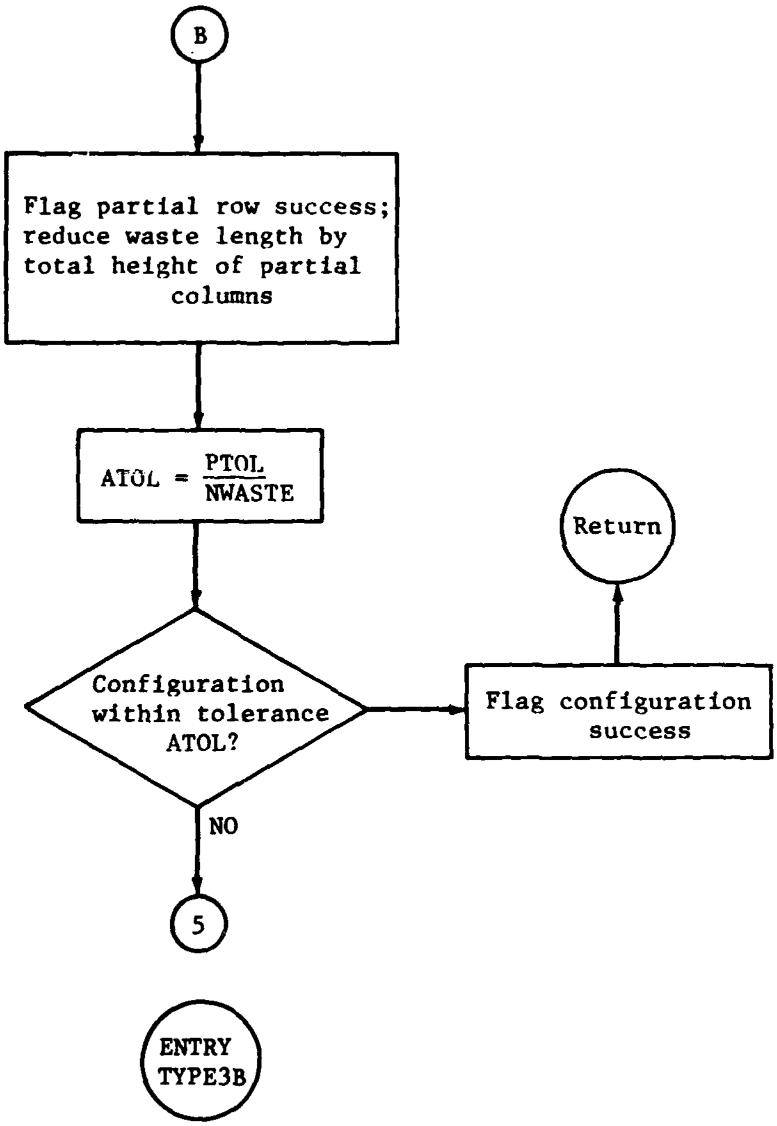


Figure 20 (Continued)



Same Flowchart as TYPE3A Except Rows and Columns are Interchanged.

Figure 21 - Subroutine TYPE4, Flowchart

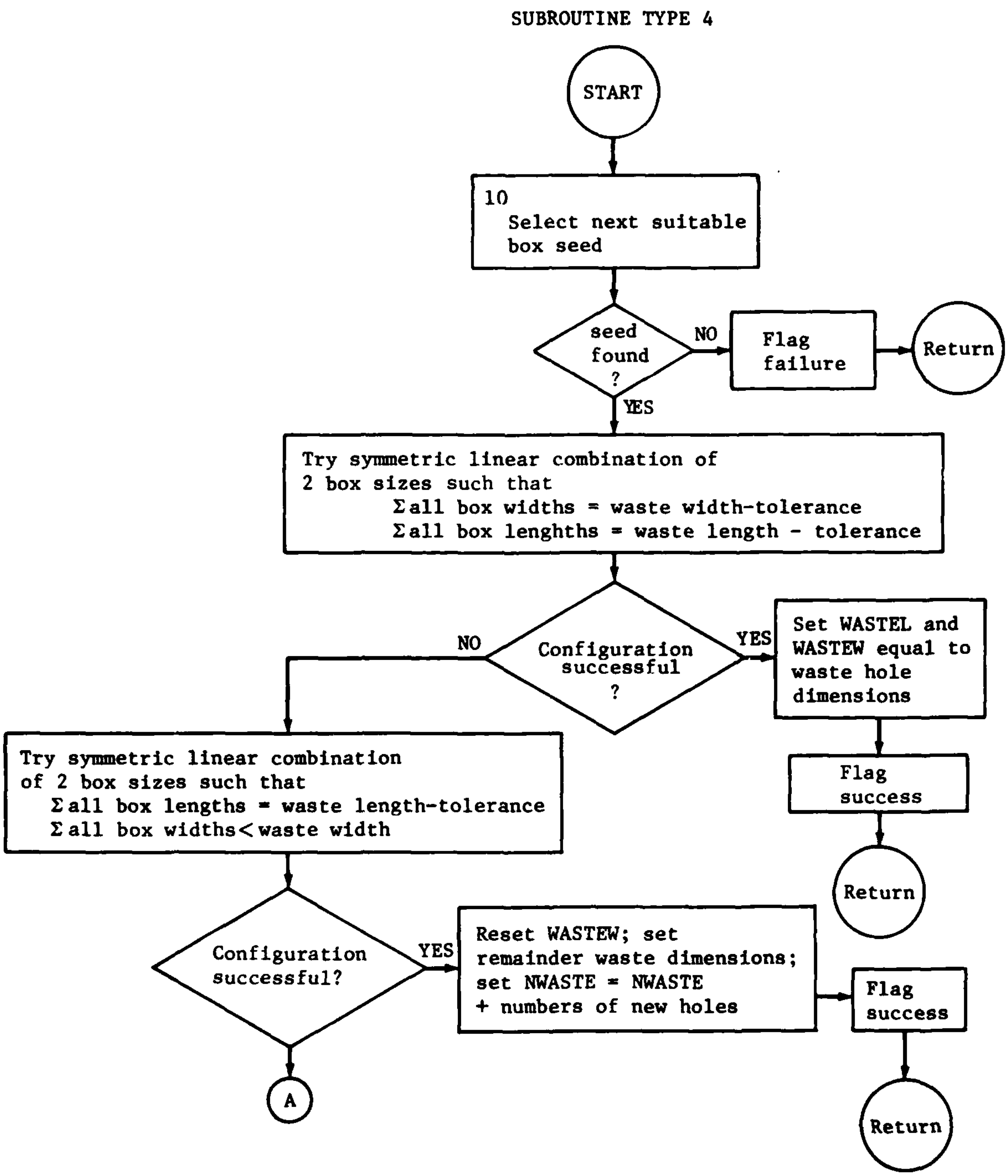


Figure 21 (Continued)

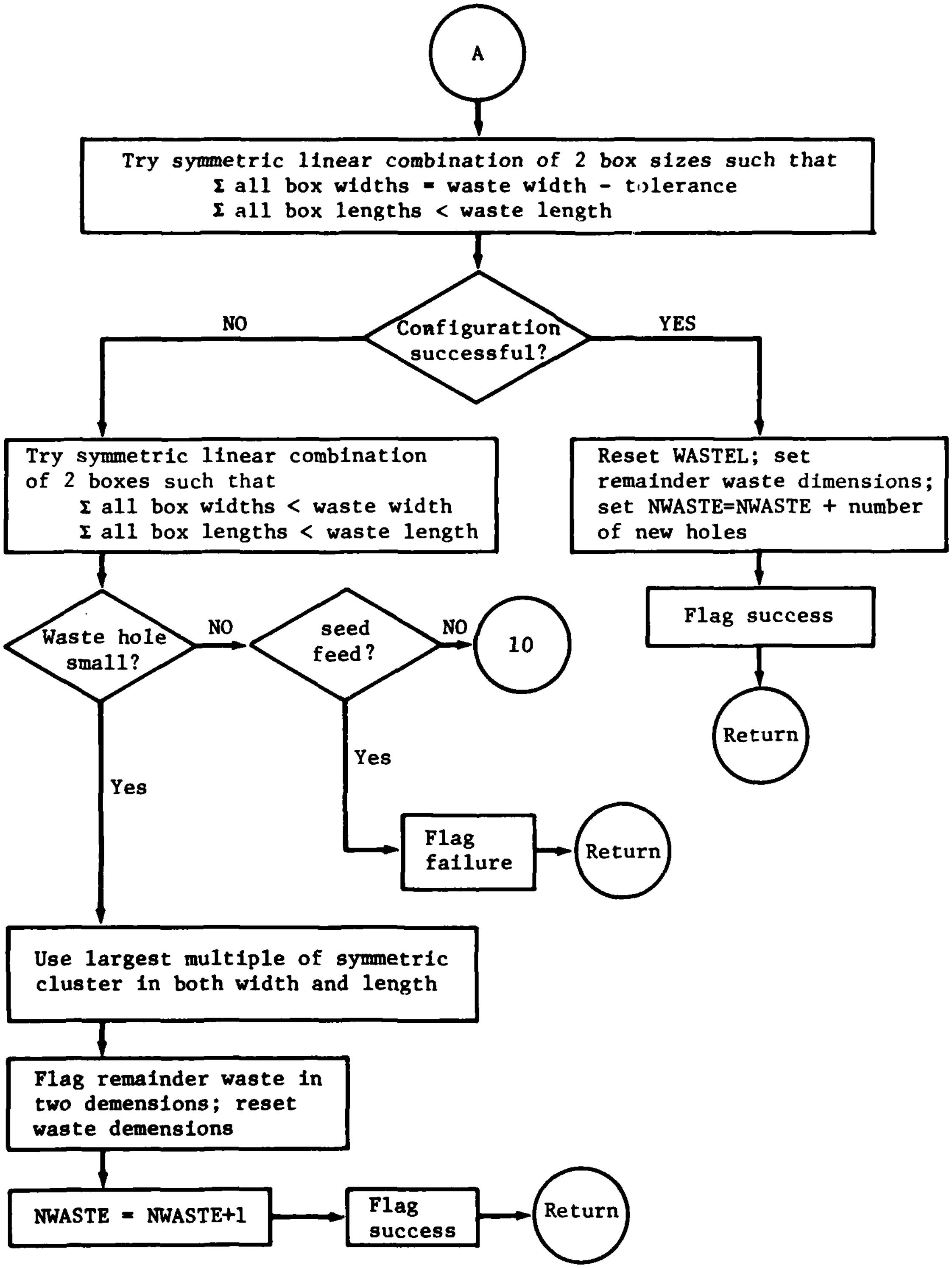


Figure 22 - Subroutine STACKS Flowchart

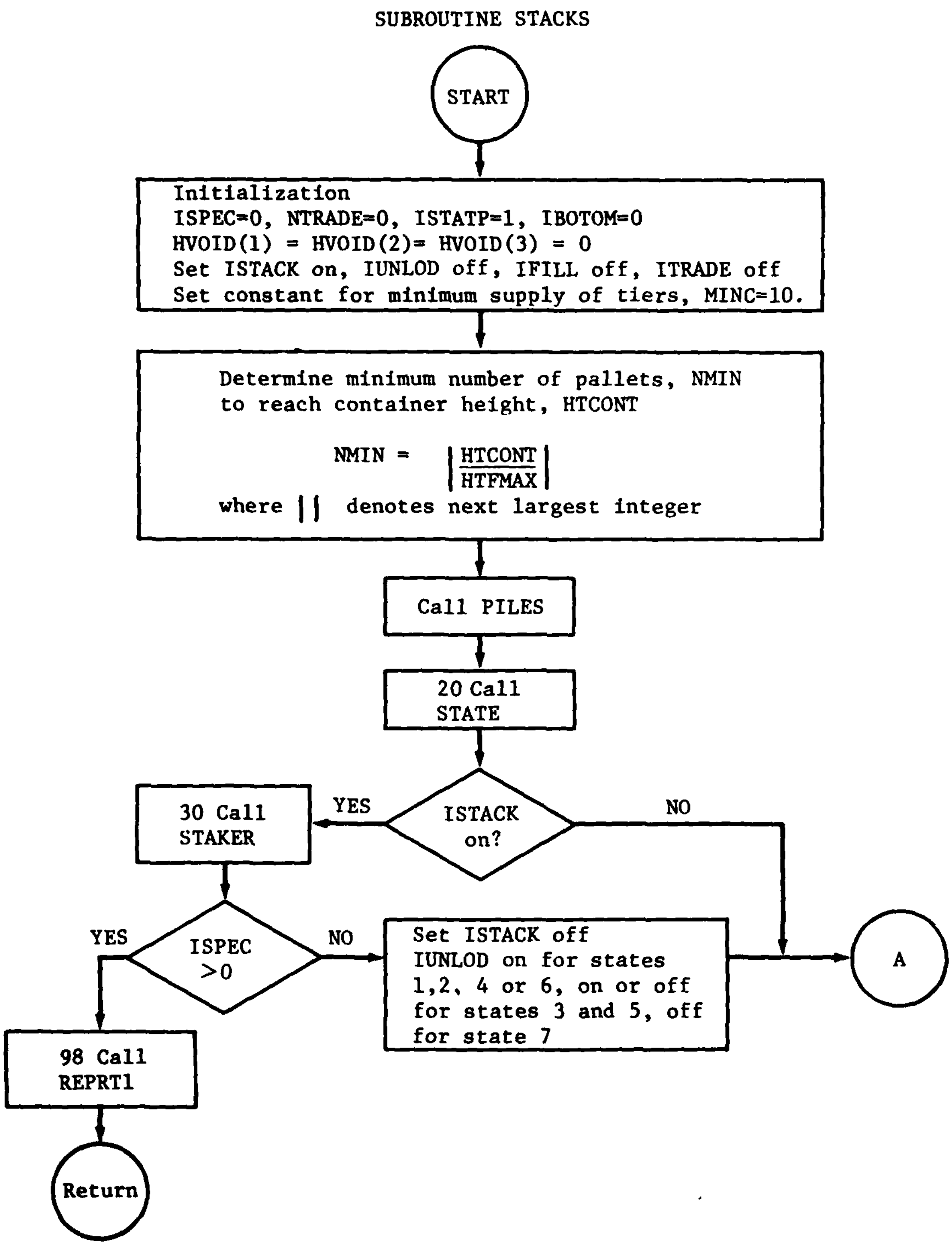


Figure 22 (Continued)

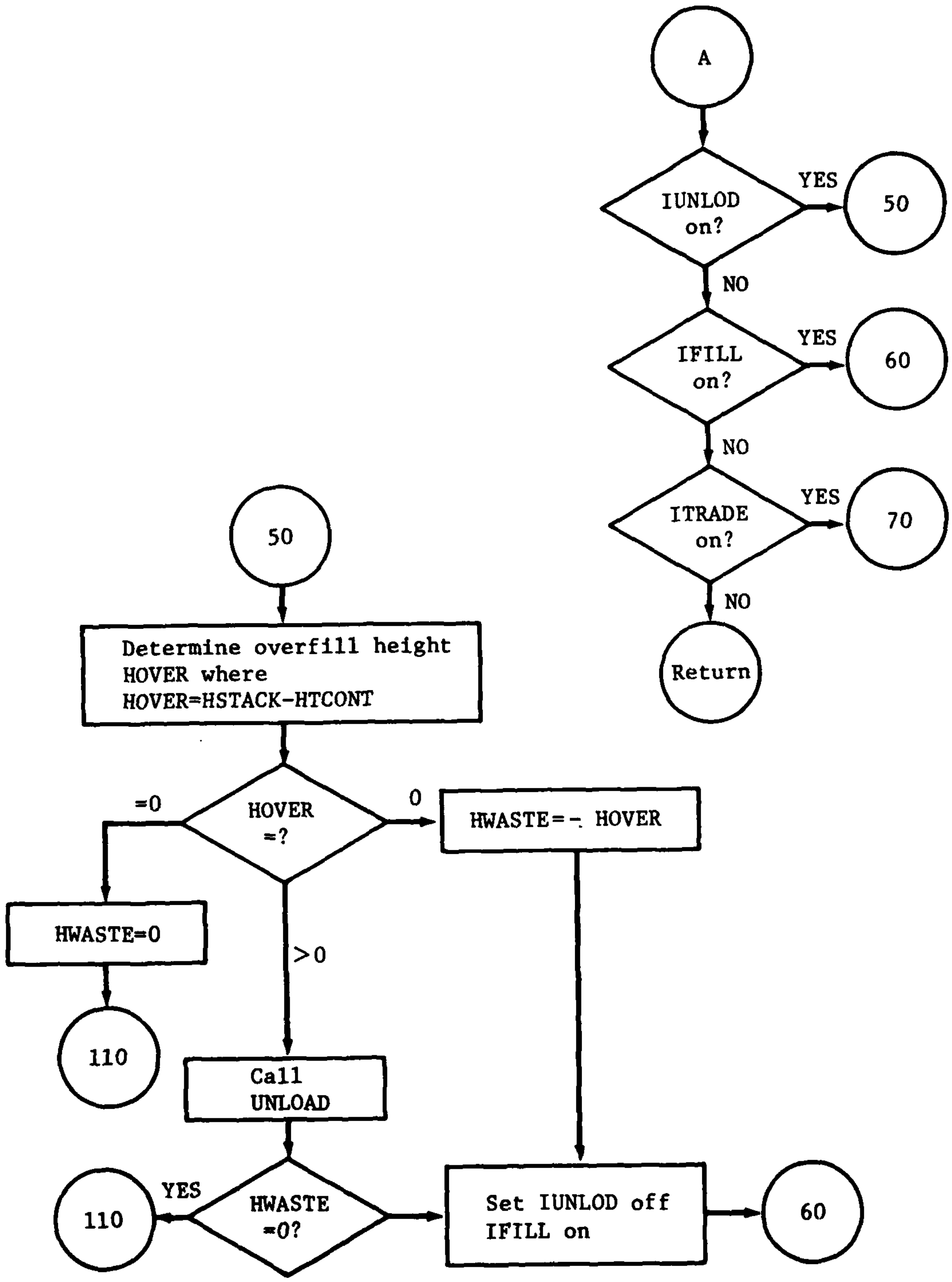


Figure 22 (Continued)

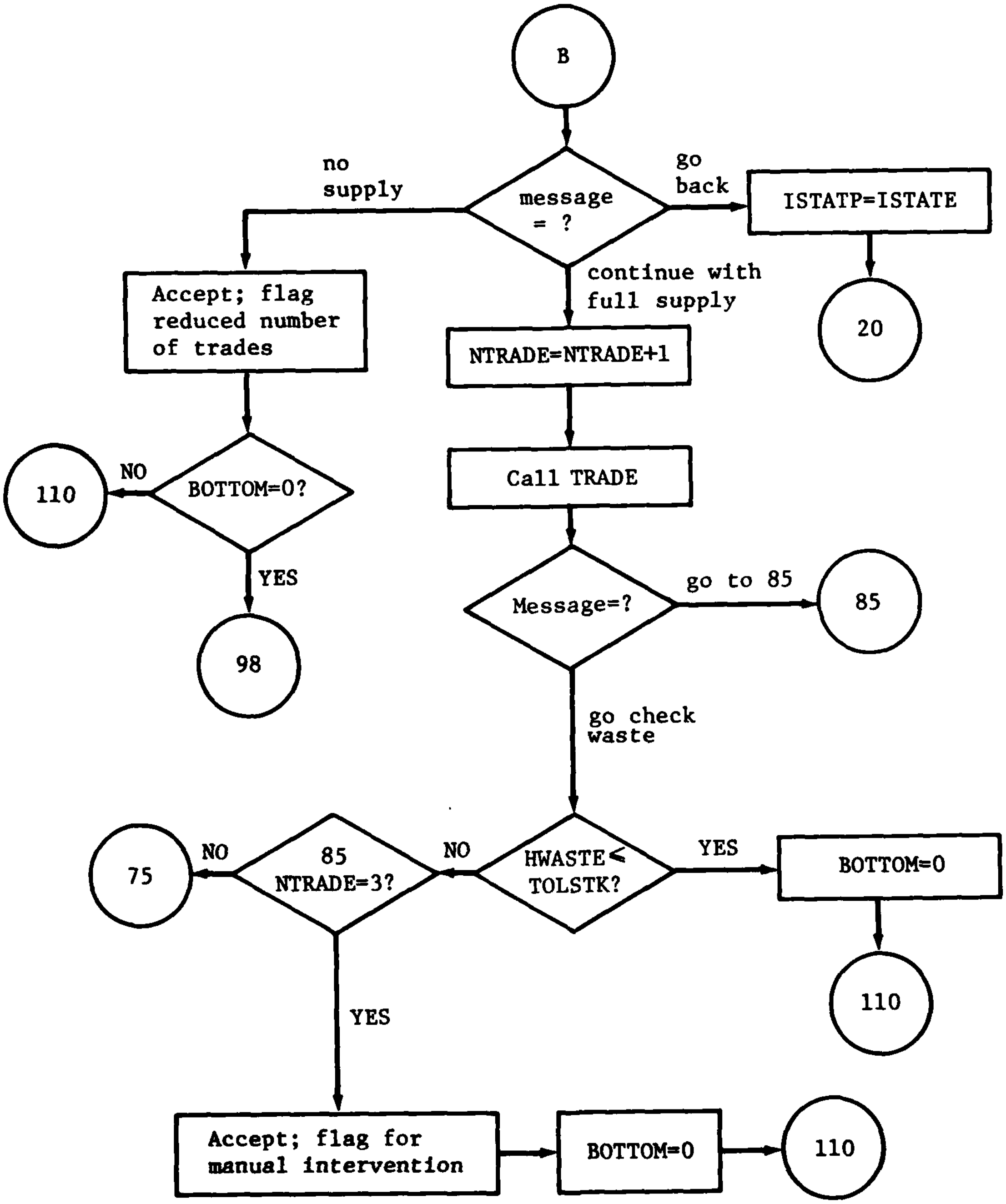


Figure 22 (Continued)

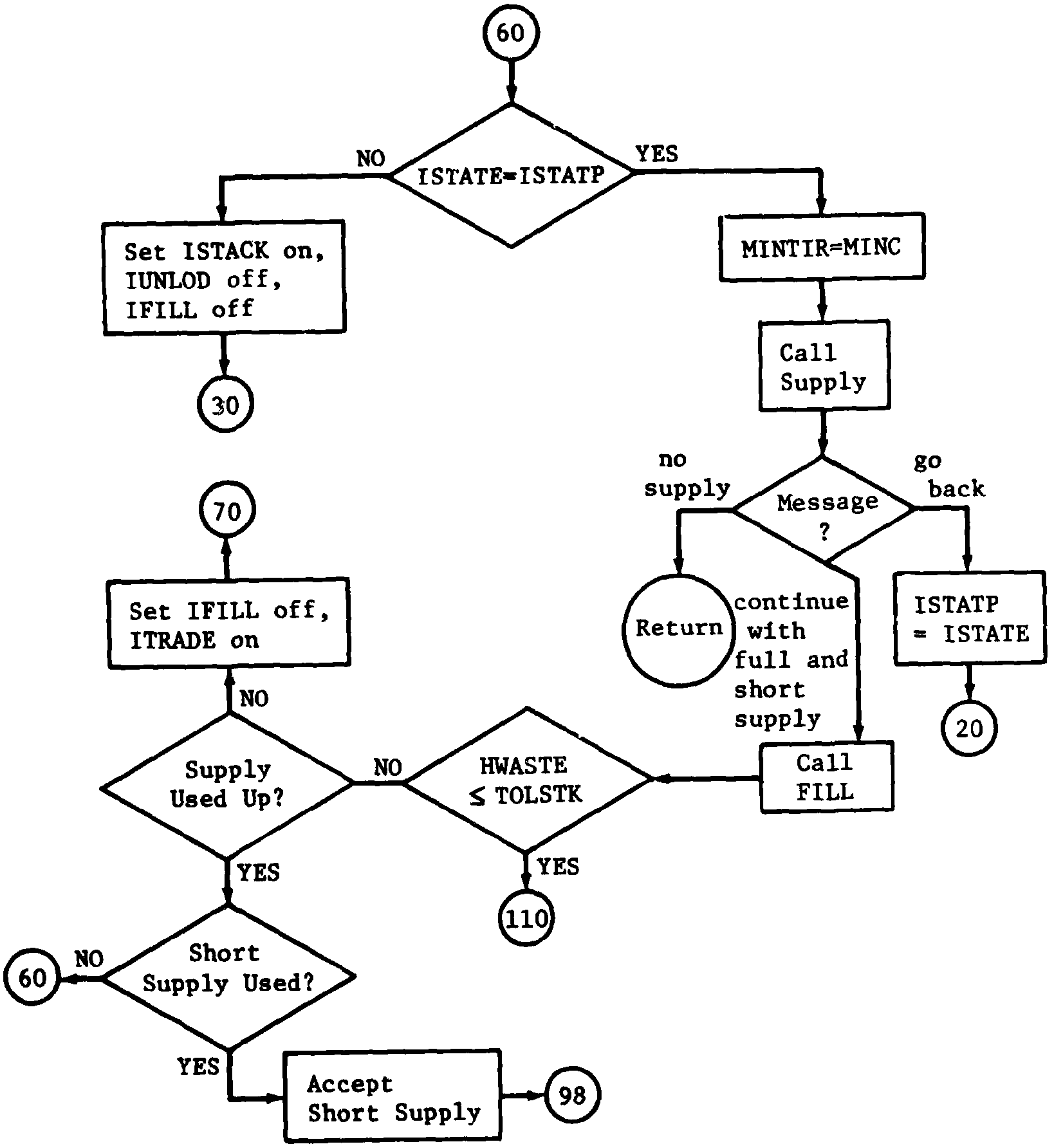
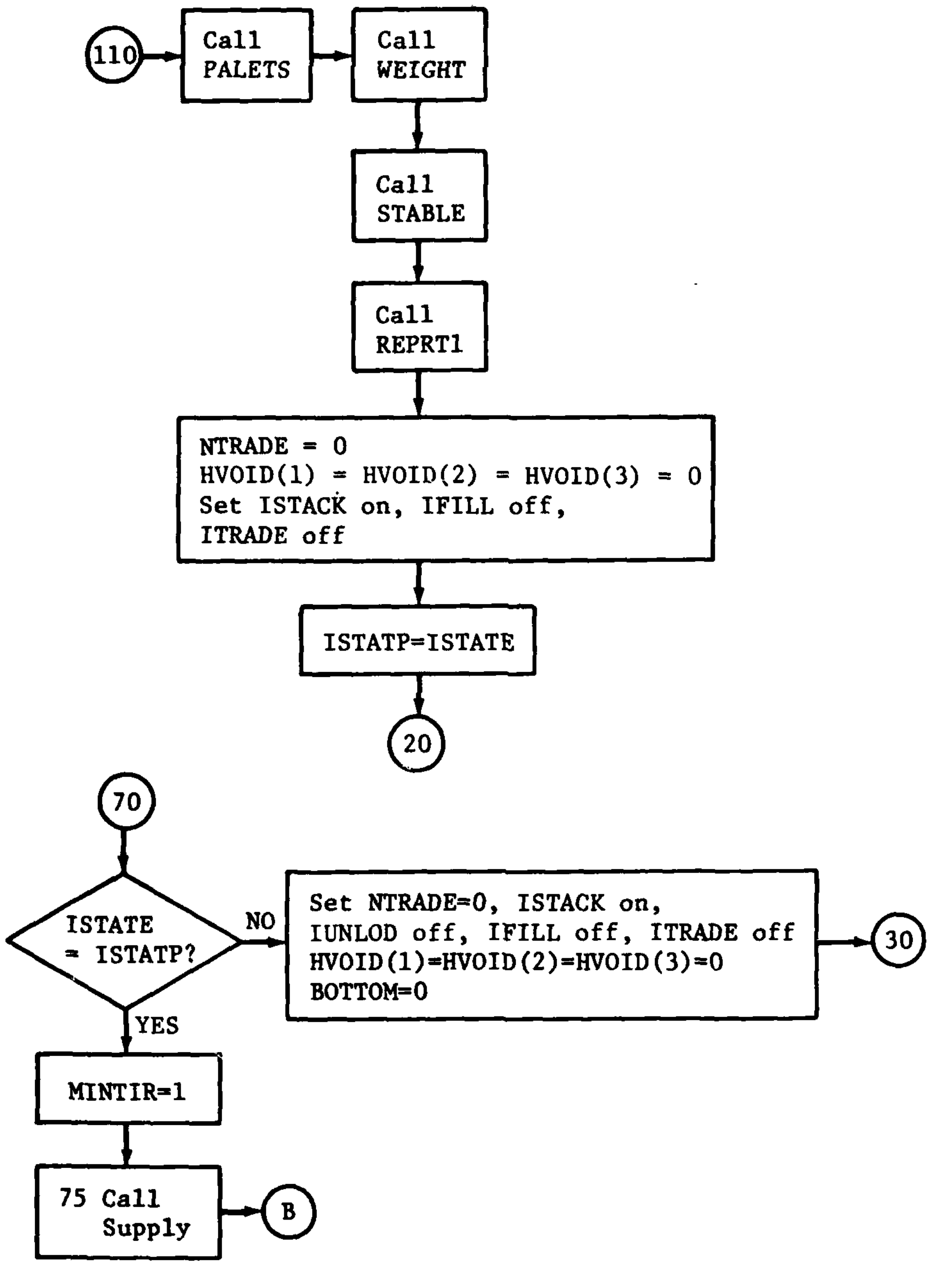


Figure 22 (Continued)



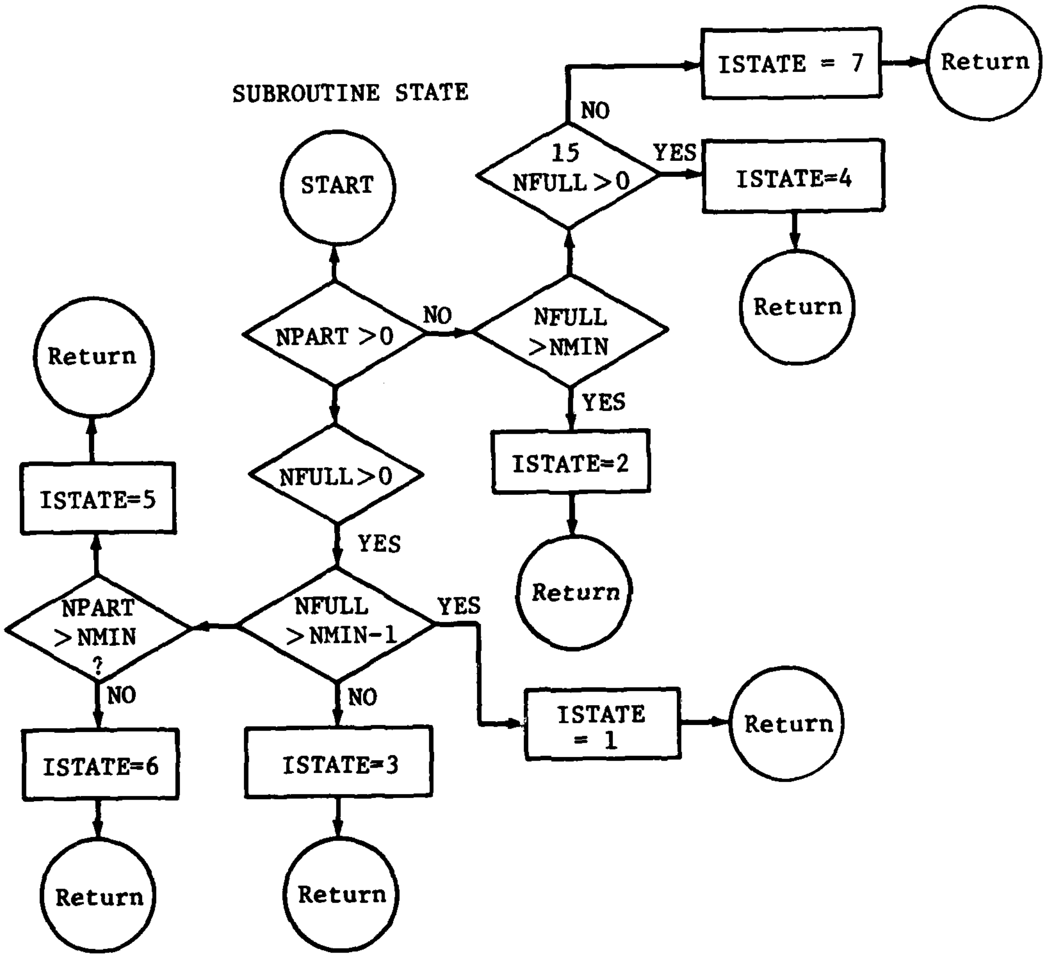


Figure 23 - Subroutine STATE, Flowchart

Figure 24 - Subroutine PILES, Flowchart

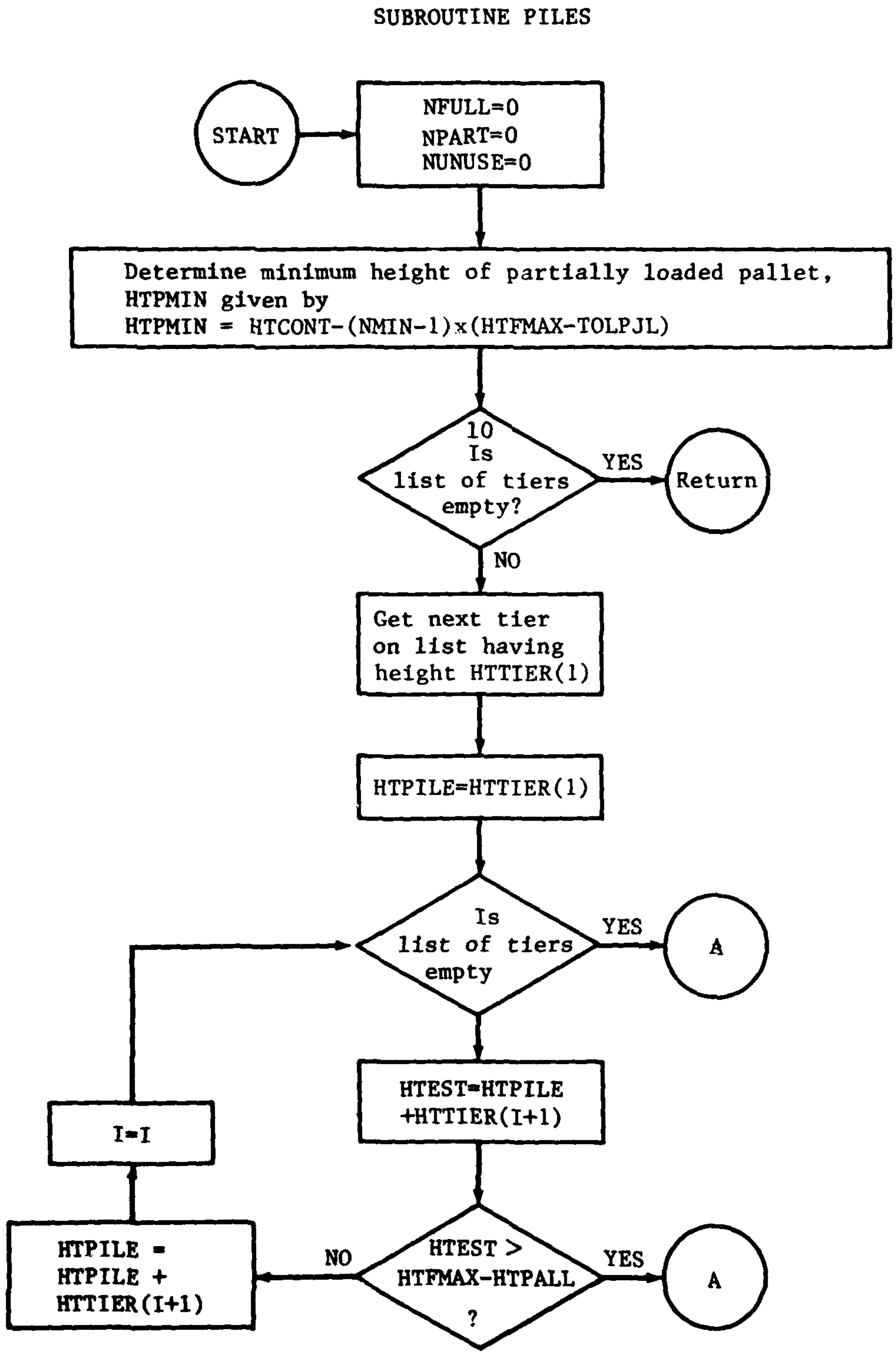


Figure 24 (Continued)

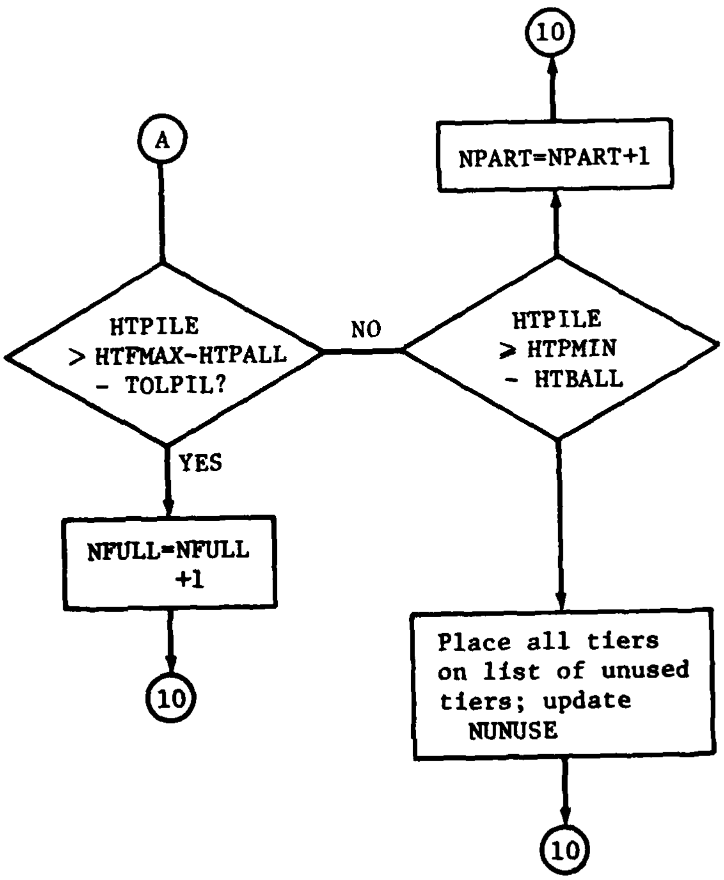


Figure 25 - Subroutine PALLETS Flowchart

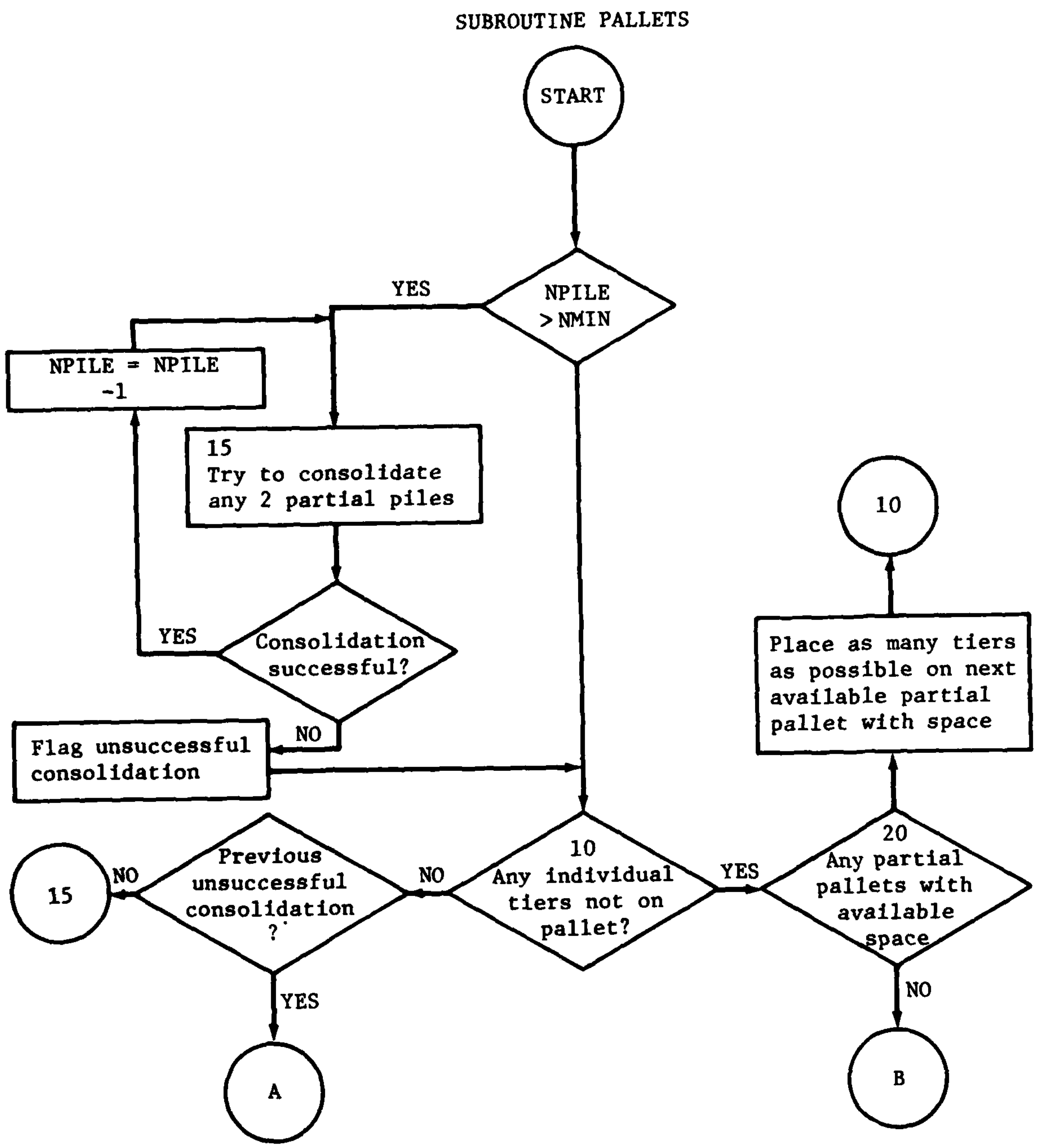


Figure 25 (Continued)

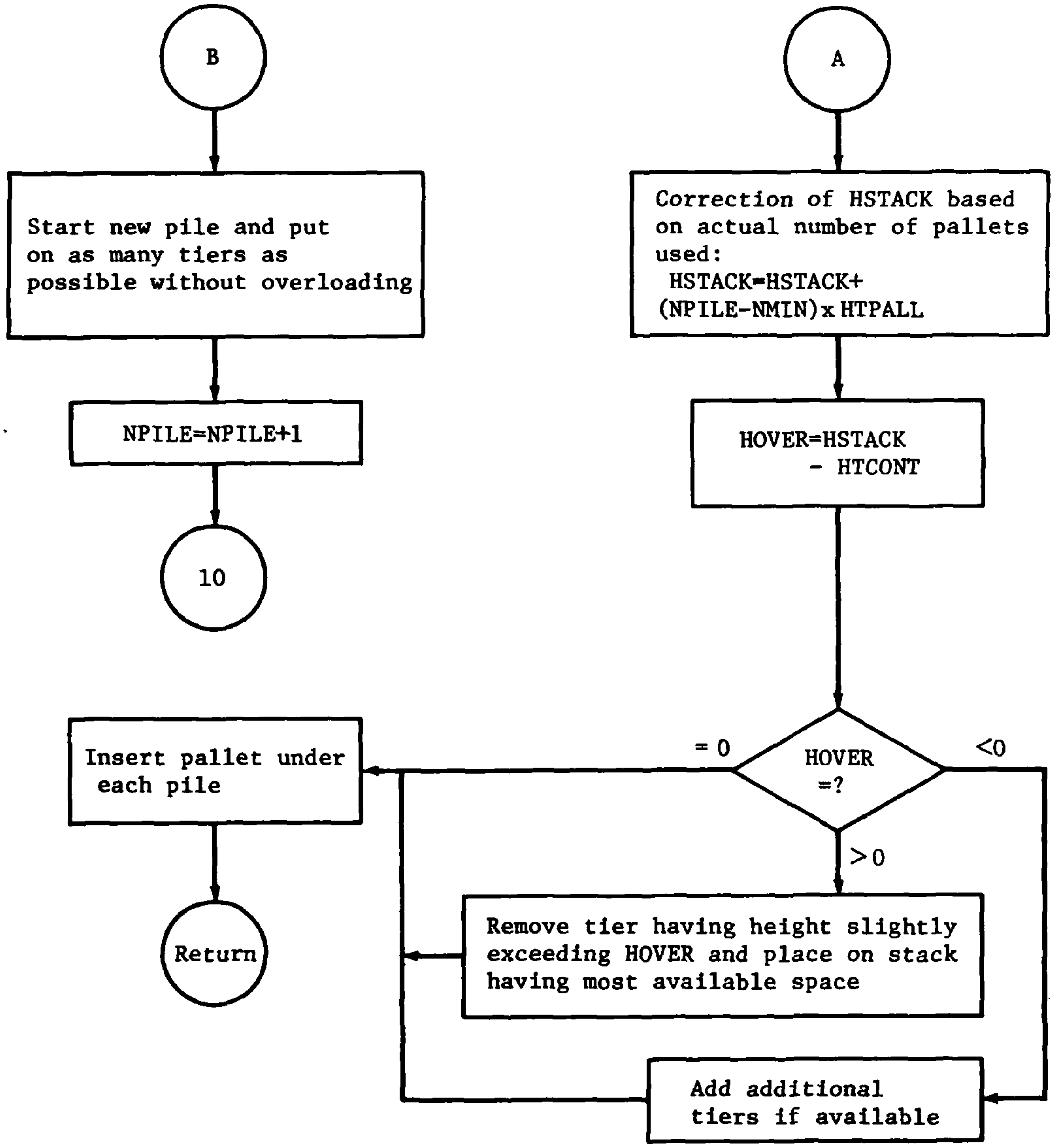
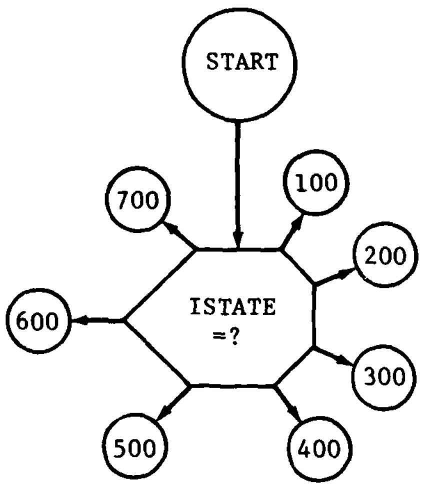


Figure 26 - Subroutine STAKER, Flowchart

SUBROUTINE STAKER



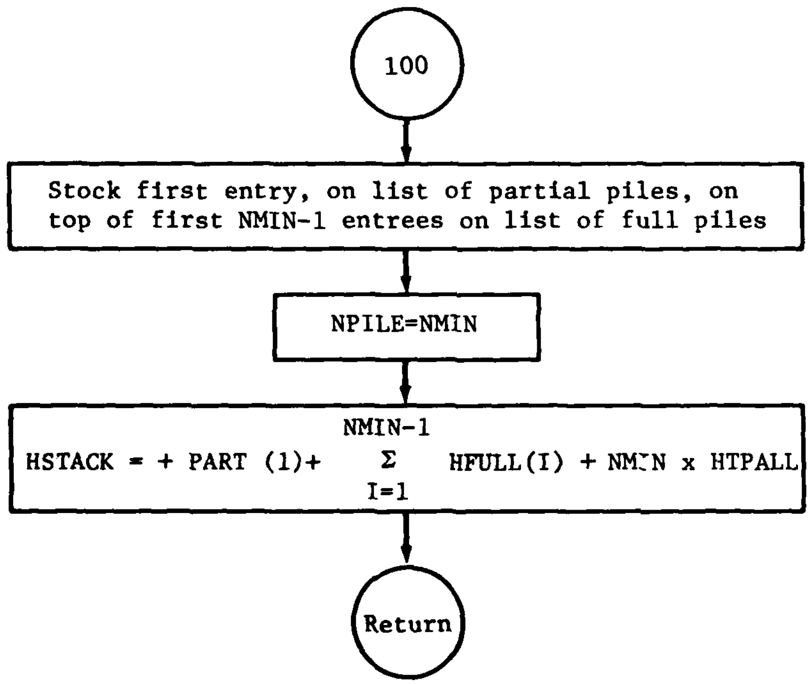


Figure 26 (Continued)

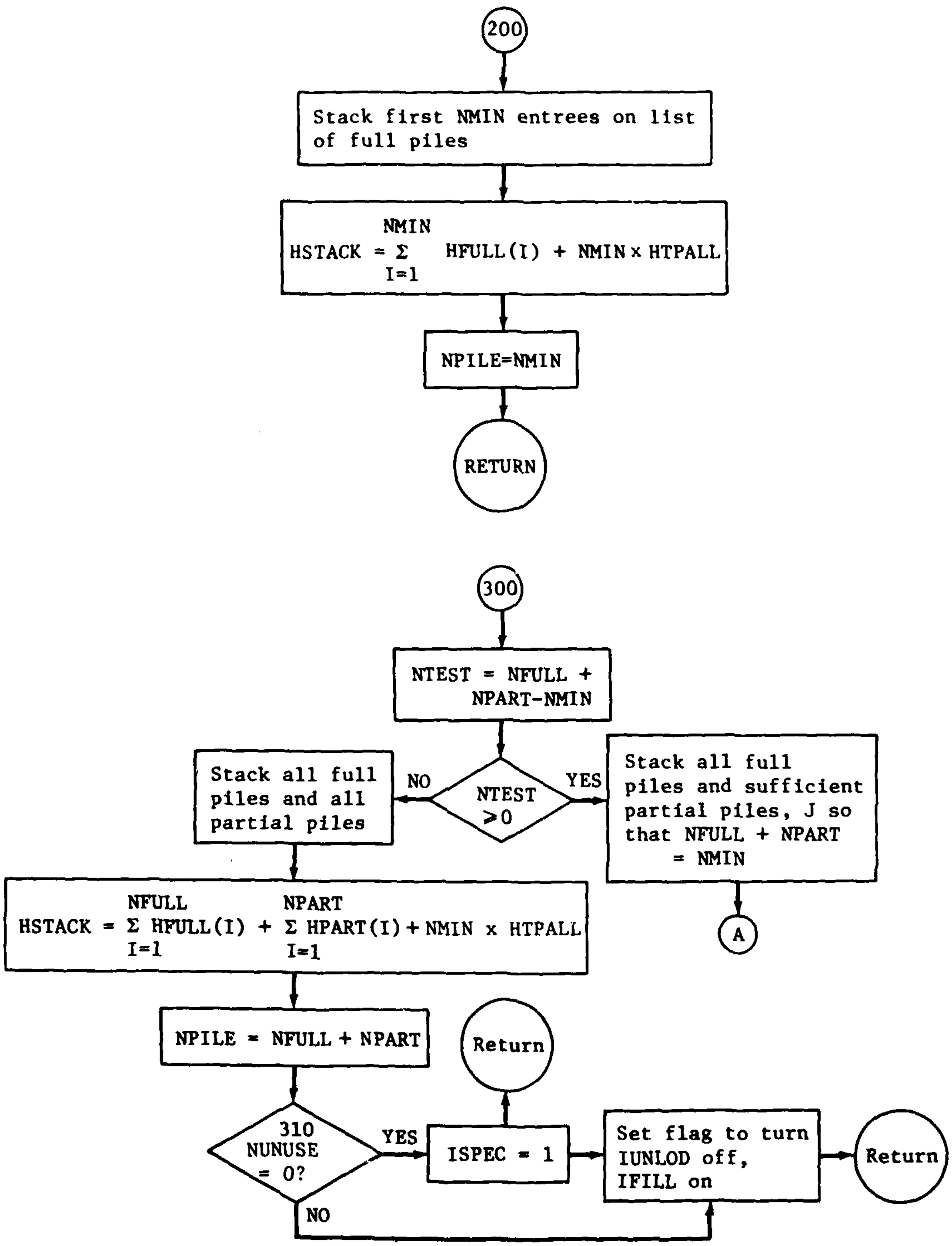


Figure 26 (Continued)

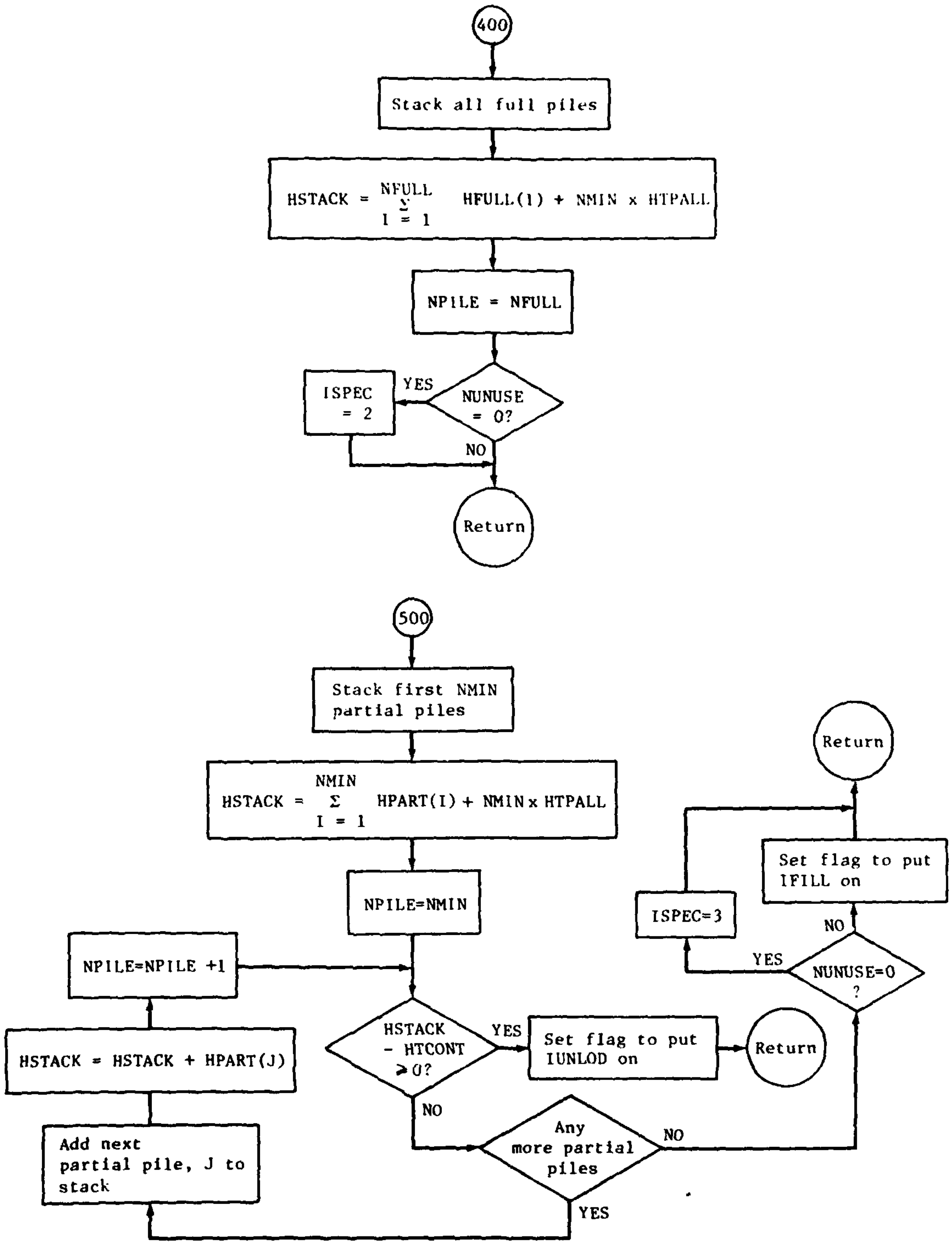


Figure 26 (Continued)

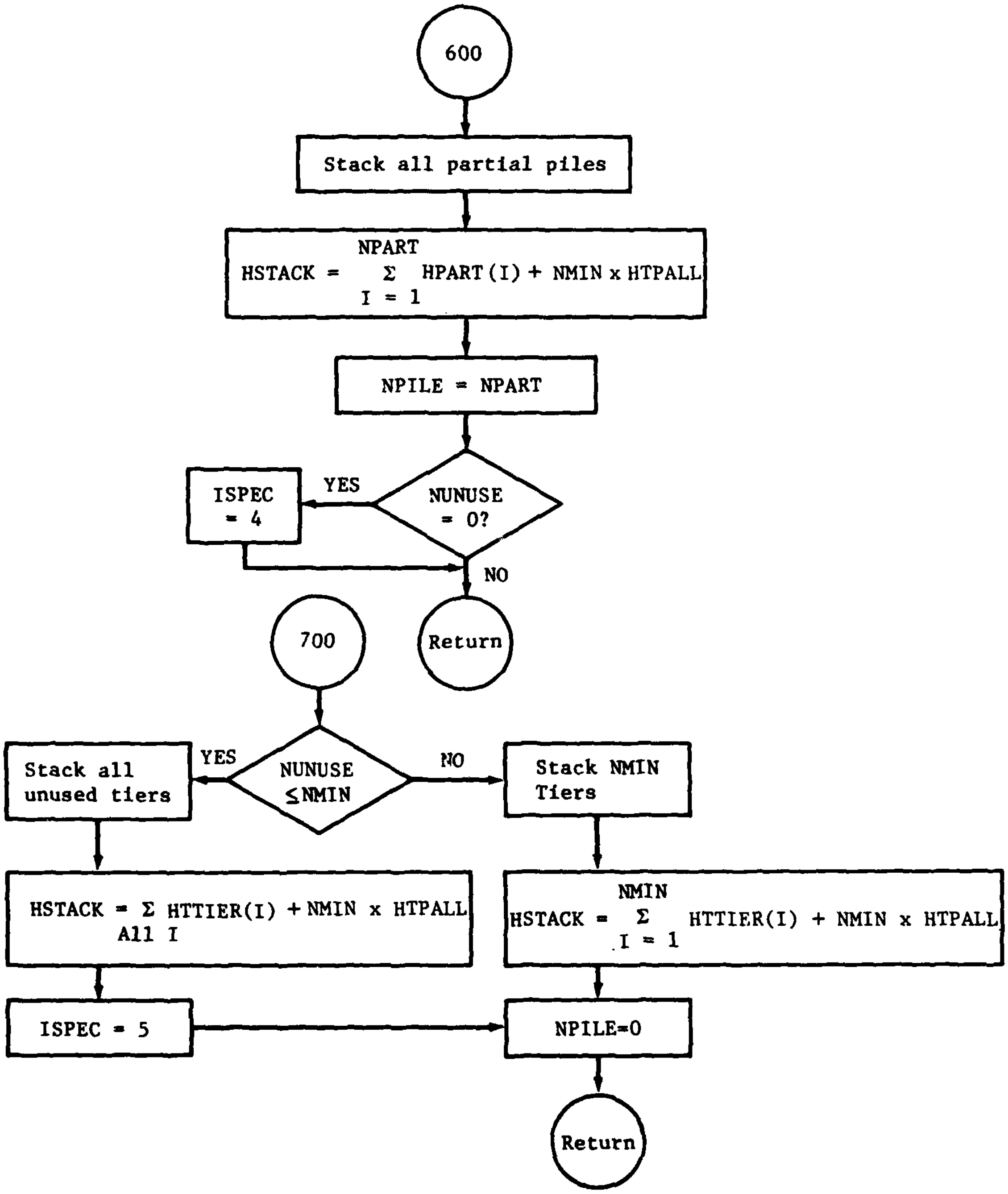
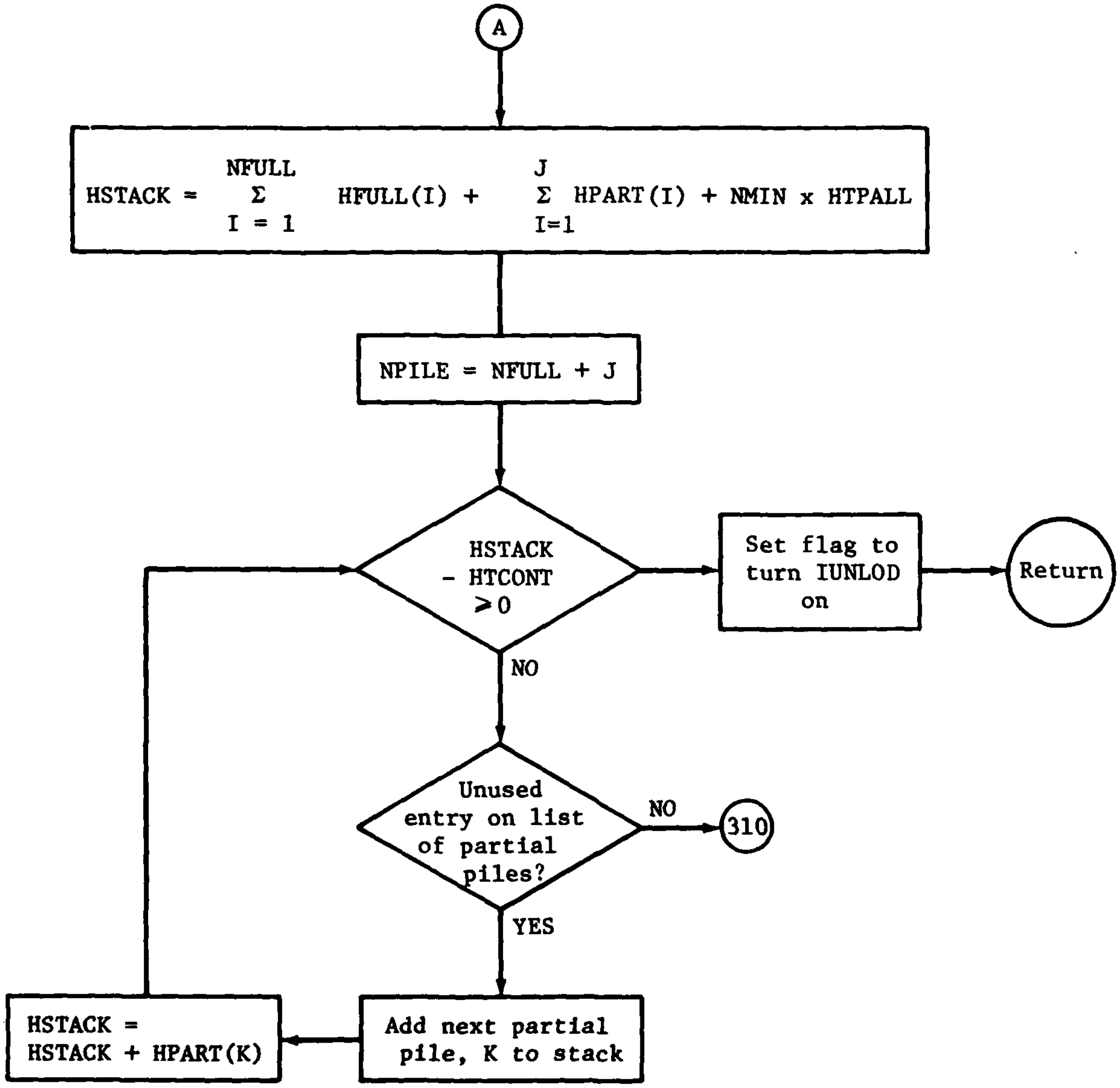


Figure 26 (Continued)



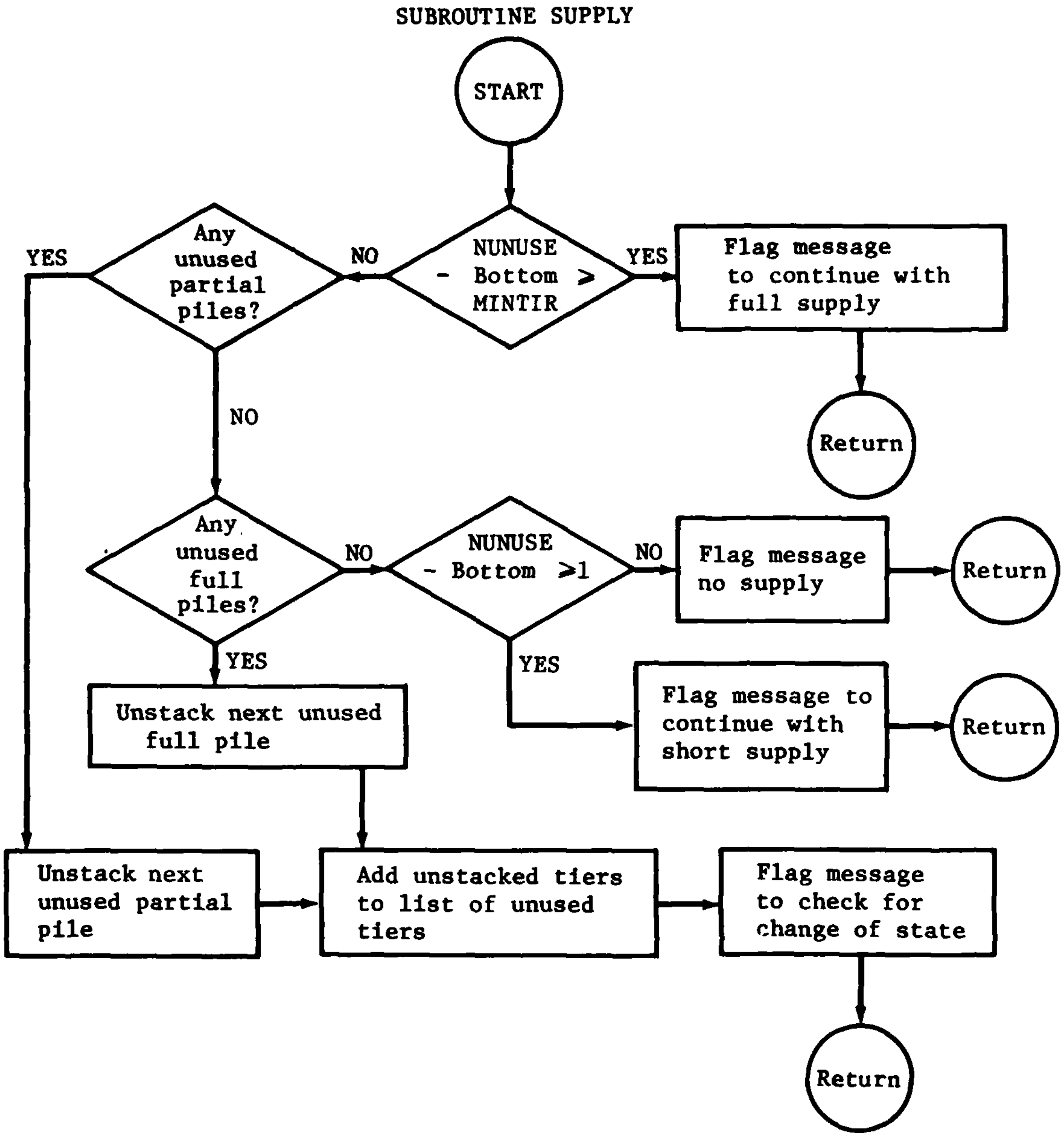


Figure 27 - Subroutine SUPPLY, Flowchart

SUBROUTINE UNLOAD

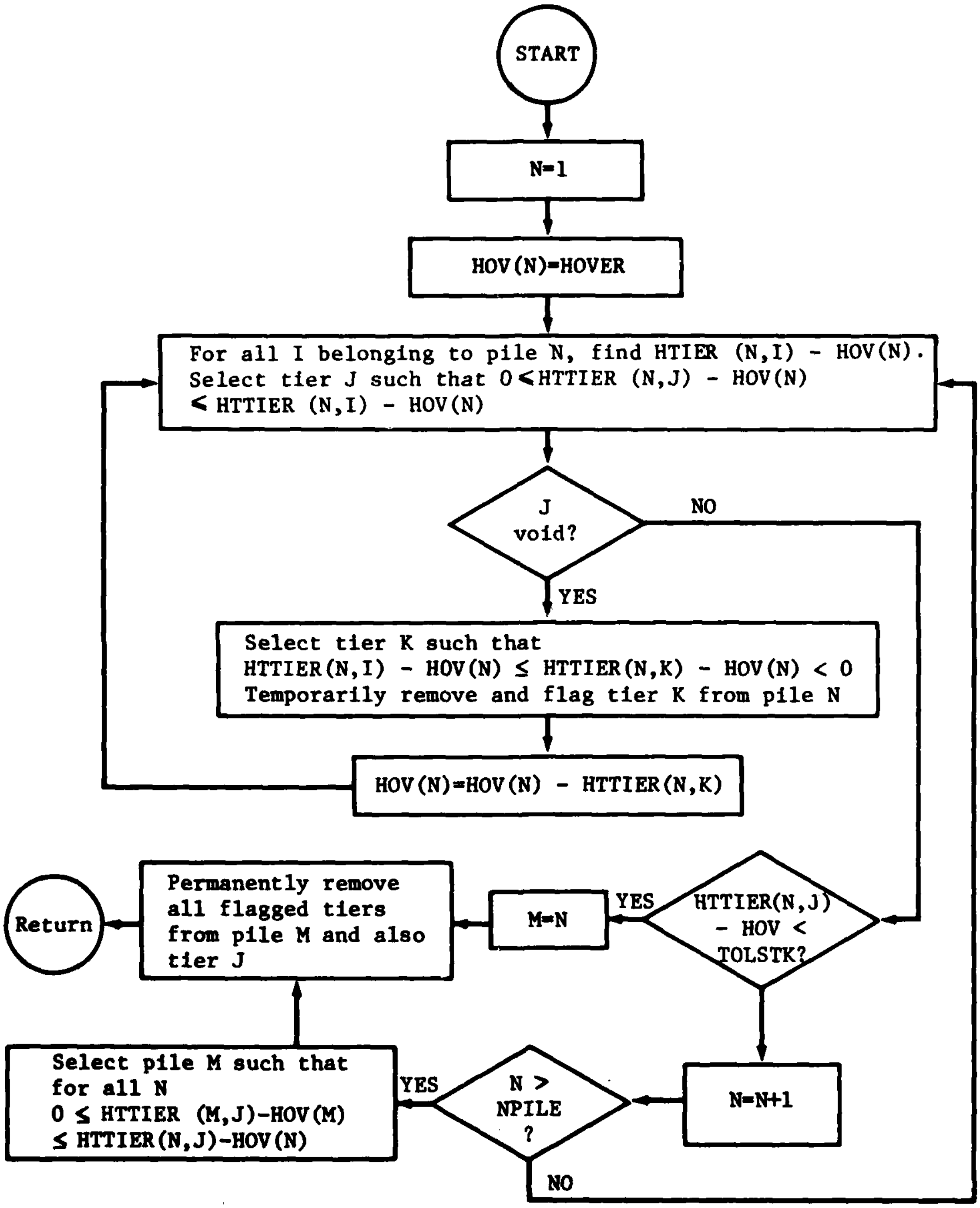


Figure 28 - Subroutine UNLOAD, Flowchart

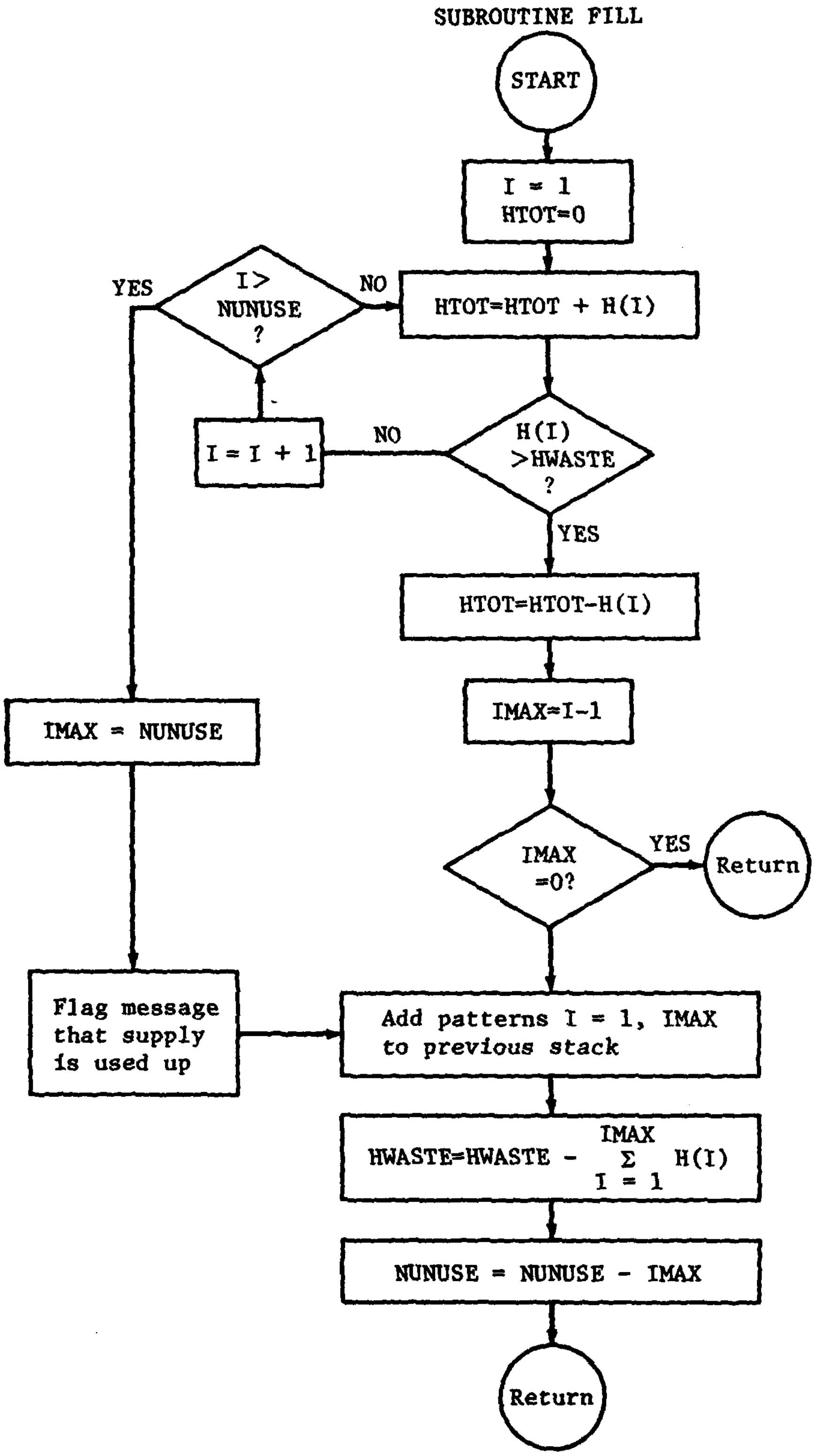


Figure 29 - Subroutine FILL, Flowchart

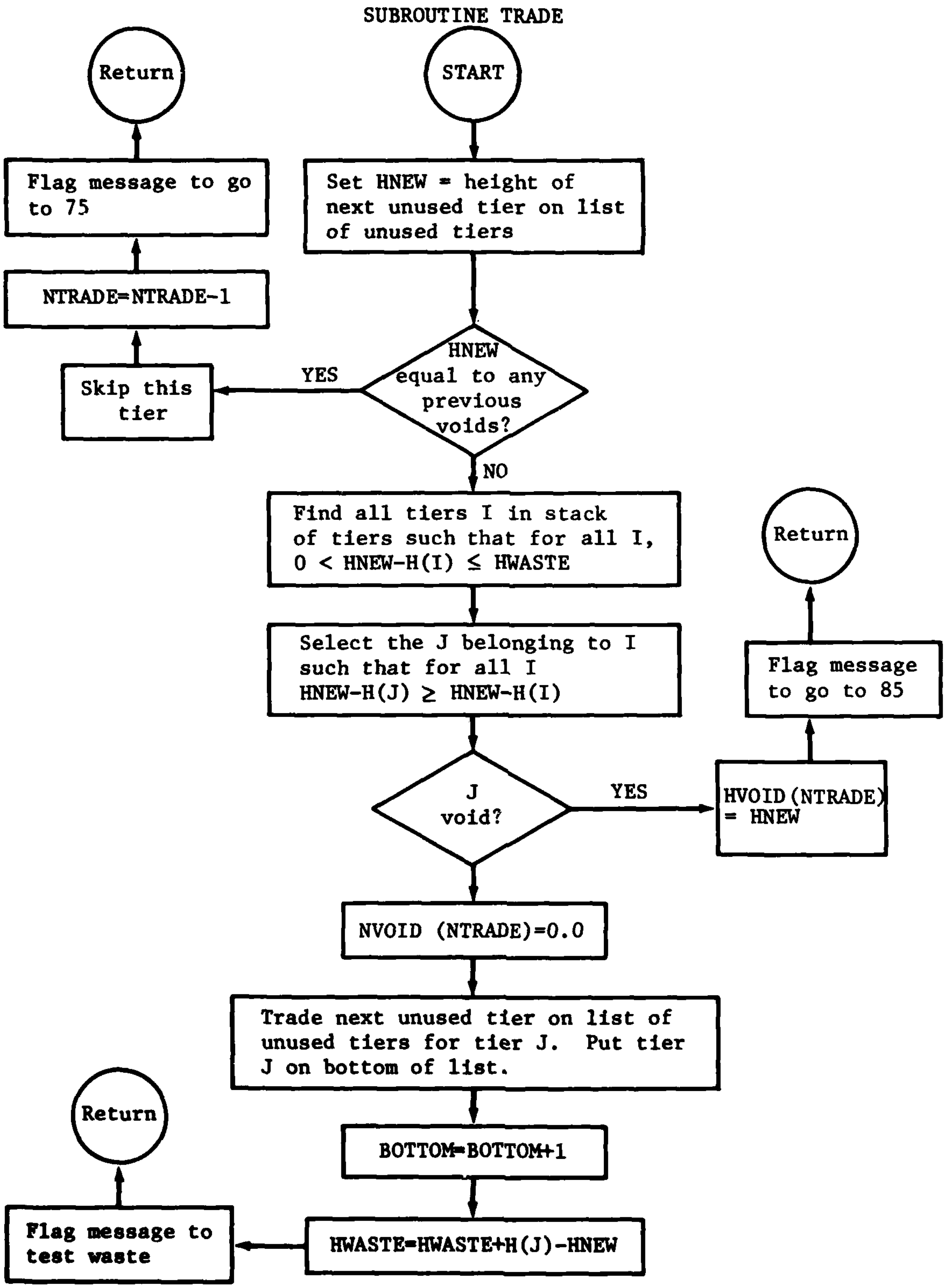


Figure 30 - Subroutine TRADE, Flowchart

SUBROUTINE CNTNER

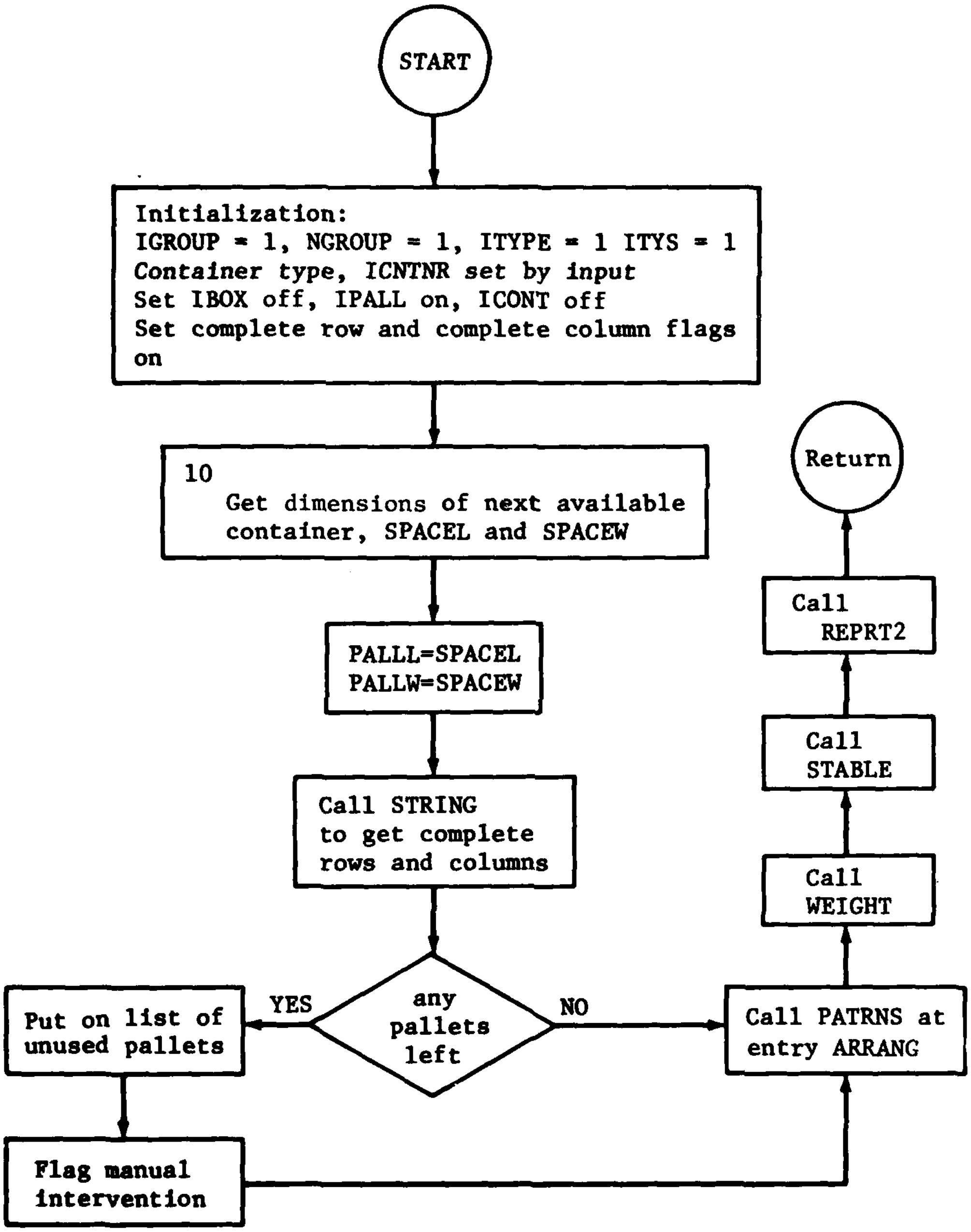


Figure 31 - Subroutine CNTNER, Flowchart

Figure 32 - Subroutine BARGE, Flowchart

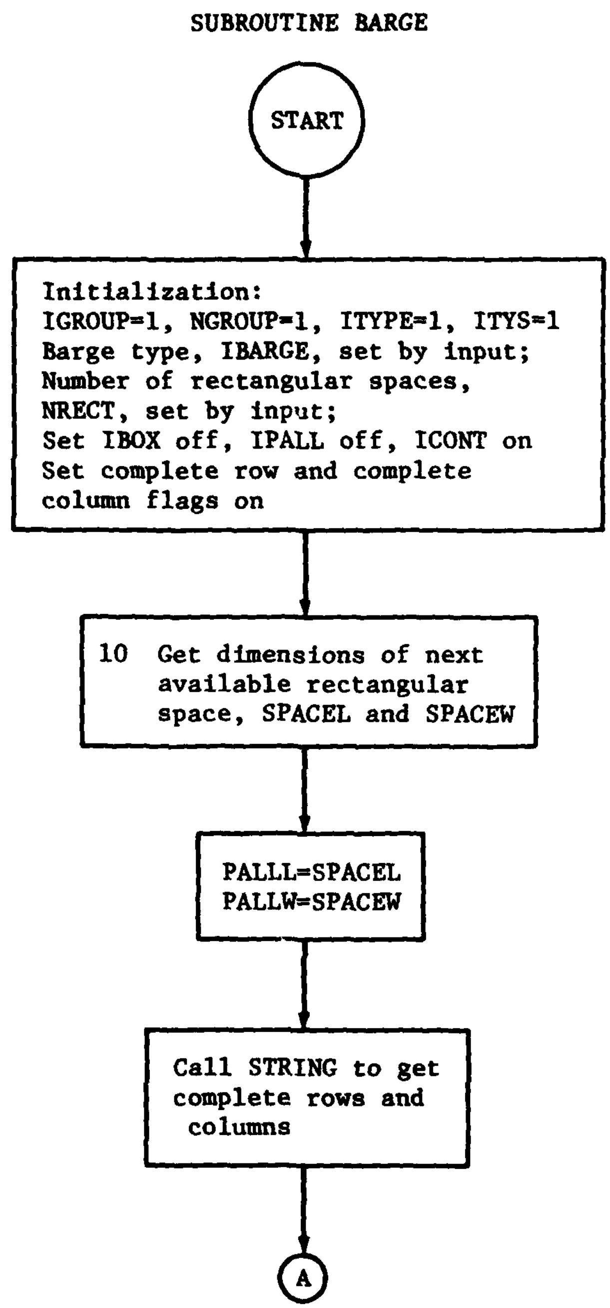
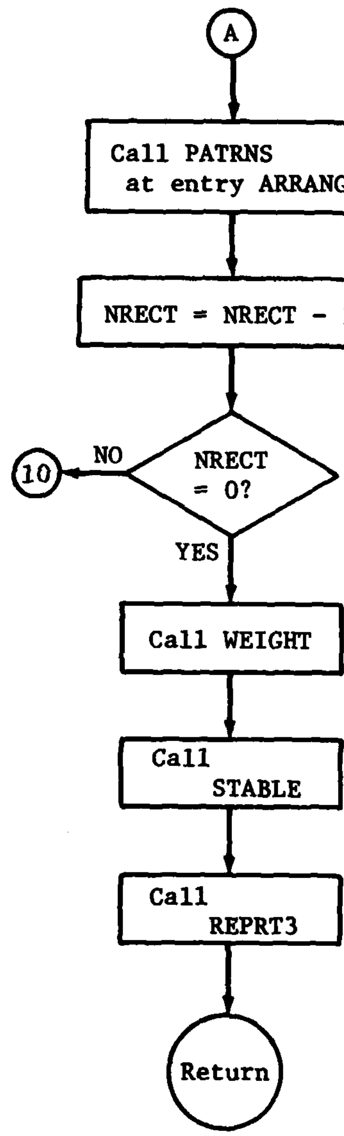


Figure 32 (Continued)



INITIAL DISTRIBUTION

Copies	
2	DLSIE
1	MSC, Code 09
2	MSDEC
10	Mr. Gerald Gerstel Hq. U.S. Army Avionics R&D Activity Attn: DAVAA-SS Ft. Monmouth, NJ 07703
1	NAVFAC, M. Essoglou
1	FMSO, Code 93
7	NAVSUP 1 SUP 03 4 SUP 04A 1 SUP 043 1 SUP 05
2	NAVPGSCOL
1	NSC CHASN, Code 400
1	NSC OAKLAND, Cdr Nelson
1	NSC PEARL HARBOR
1	NSC PUGET, Code 43
1	NSC SAN DIEGO
12	DTIC

CENTER DISTRIBUTION

Copies	Code	Name
1	1800	G.H. Gleissner
2	1809.3	D. Harris
1	185	T. Corin
20	187	M. Zuhkoff
1	93	L. Marsh
10	5211.1	Reports Distribution
1	522.1	Unclassified Lib (C) (m)
1	522.2	Unclassified Lib (A)

LMED

-8